PROJETO FINAL DE GRADUAÇÃO

Blockchain-based Authentication Protocol for Internet of Medical Things (IoMT)

Guilherme Luís Rocha

Curso Superior de Engenharia de Redes de Comunicação

DEPARTAMENTO DE ENGENHARIA ELÉTRICA
FACULDADE DE TECNOLOGIA
UNIVERSIDADE DE BRASÍLIA

UNIVERSIDADE DE BRASÍLIA
Faculdade de Tecnologia

PROJETO FINAL DE GRADUAÇÃO

Blockchain-based Authentication Protocol for Internet of Medical Things (IoMT)

Guilherme Luís Rocha

Projeto Final de Graduação submetida ao Departamento de Engenharia Elétrica como requisito parcial para obtenção do grau de Bacharel em Engenharia de Redes de Comunicação

FICHA CATALOGRÁFICA

ROCHA, G.L.R.
Blockchain-based Authentication Protocol for Internet of Medical Things (IoMT) [Distrito Federal] 2023.
xvi, 58 p., 210 x 297 mm (ENE/FT/UnB, Bacharel, Engenharia de Redes de Comunicação, 2023).
Projeto Final de Graduação - Universidade de Brasília, Faculdade de Tecnologia.
Departamento de Engenharia Elétrica

1. Blockchain	2. Autenticação e Acordo de Chaves
3. Internet das Coisas (IoT)	4. Segurança da Informação
I. ENE/FT/UnB	II. Título (série)

REFERÊNCIA BIBLIOGRÁFICA

ROCHA, G.L.R. (2023). *Blockchain-based Authentication Protocol for Internet of Medical Things (IoMT)*. Projeto Final de Graduação, Departamento de Engenharia Elétrica, Universidade de Brasília, Brasília, DF, 58 p.

CESSÃO DE DIREITOS

AUTOR: Guilherme Luís Rocha
TÍTULO: Blockchain-based Authentication Protocol for Internet of Medical Things (IoMT).
GRAU: Bacharel em Engenharia de Redes de Comunicação ANO: 2023

Guilherme Luís Rocha
Depto. de Engenharia Elétrica (ENE) - FT
Universidade de Brasília (UnB)
Campus Darcy Ribeiro
CEP: 70919-970 - Brasília-DF - Brasil

Dedico este trabalho à minha família e amigos, pelo apoio e camaradagem pelos últimos anos para chegar até onde estou hoje, através de todas as tribulações. Acima de tudo, dedico este trabalho a Deus, que me preparou e me deu a oportunidade de trabalhar neste projeto.

AGRADECIMENTOS

Primeiramente agradeço a Deus por mais uma vitória nessa vida, e pela trajetória que ele montou para eu chegar aonde estou hoje, cheio de diversas bençãos. Agradeço a minha família, que por todas as adversidades se manteve unida, e me deu o apoio, que em todos os momentos que não acreditava conseguir, esteve comigo. Agradeço a meus colegas do curso de Engenharia de Redes de Comunicação, que juntos comigo, atravessamos e conquistamos diversos obstáculos. Agradeço por fim a todos os meus professores, que me conduziram nesse caminho de aprendizado, em especial ao meu orientador Paulo Gondim, que possibilitou a execução deste trabalho, por meio de seus ensinamentos.

RESUMO

Blockchain é visto como uma tecnologia promissora para propulsionar as comunicações de redes heterogêneas, sendo um alvo de muitos estudos para relações 5G e IoT (Internet of Things). É imaginado que tecnologias na corrente seriam capazes de propulsionar a integridade, imutabilidade, privacidade, contabilização e segurnaça necessárias para manter essas redes no mundo digital com ameaças cibernéticas cada vez mais potentes e adaptáveis.

A segurança é um aspecto crucial para redes IoT, e por complemento, redes IoMT (Internet of Medical Things), que devido a sua posição como um ambiente de recursos restritos, requer novas técnicas para ser capaz de se manter em dia com as novas ameaças no cenário. Deste modo, a autenticação e acordo de chaves (AKA, do inglês Authentication and Key Agreement). Entretanto, os protocolos de autenticação utilizados nas redes tradicionais (como os protocolos EPS-AKA e EAP-AKA) não estão adaptados para essa nova realidade, com muitos autores demonstrando preocupação com seu uso futuro.

Este trabalho apresenta o projeto e a avaliação de um protocolo de autenticação, desenvolvido para o cenário de dispositivos IoMT, usando redes 5G.

A metodologia para o desenvolvimento do protocolo considerou, como passo inicial, uma revisão da literatura, buscando identificar protocolos que tenham sido empregados, de forma específica, em cada cenário considerado. Em seguida, a proposta da arquitetura a ser utilizada, com os objetivos a serem alcançados, bem como propriedades de segurança, possíveis ataques e vulnerabilidades do modelo. É então proposto um novo protocolo de autenticação para o cenário.

Uma análise e construção dos mecanismos de blockchain são apresentados, e posteriormente é trabalhado ataques, como Man-In-The-Middle (MITM), ataques de identidade, ataques internos, malware, ataques baseados em sessão, entre outros.

Após a descrição do protocolo, esta dissertação apresenta comparações em relação a propriedades de segurança entre o protocolo proposto e alguns de seus respectivos trabalhos relacionados. Uma comparação envolvendo custos de computação, de comunicação e de armazenamento é então realizada. Os resultados obtidos mostram bom desempenho e robustez em segurança para o esquema proposto.

Uma validação semiformal das propriedeades de segurança do protocolo são apresentadas. Posteriormente uma verificação formal é realizada por meio da ferramenta AVISPA.

Palavras-chave:Blockchain, Internet das Coisas (IoT), Internet das Coisas Médicas (IoMT), segurança. Autenticação e Acordo de Chaves

ABSTRACT

Blockchain is seen as a promising technology for boosting heterogeneous network communications, and has been the focus of many studies on 5G and IoT (Internet of Things) relationships. It is envisioned that current technologies would be able to propel the integrity, immutability, privacy, accounting and security necessary for the maintainance of those networks in the digital world with increasingly powered and adaptable cyber threats.

Security is a crucial aspect for IoT networks, and by extension, IoMT (Internet of Medical Things) networks, which due to their position as a resource-constrained environment, require new techniques to be able to keep up with new threats in the scenario. In this way, proper implementation of Authentication and key Agreement (AKA) is required, and is found on many real life applications. However, authentication protocols used in traditional networks (such as EPS-AKA and EAP-AKA protocols) are not adapted to the new reality being offered by IoT netwoks, with many authors expressing concern about their future use.

This work addresses the design and evaluation of an authentication protocol, developed for the scenario of IoMT devices, and that uses 5G networks. The methodology for its development considered a review of the literature as an initial step for identifying the protocols that have been employed, in that scenario. The work also presents the architecture to be used, with the objectives aimed at, and security properties, possible attacks, and vulnerabilities of the model. A new authentication protocol is then proposed for the scenario.

An analysis and construction of blockchain mechanisms are presented and attacks such as Man-In-The-Middle (MiTM), identity, insider, malware, session-based ones, among others, discussed.

The security properties of the proposed protocol and of some related ones, as well as computational, communication, and storage costs are then compared. The results show good performance and security robustness of the proposed scheme.

A semi-formal validation of the protocol's security properties is presented, and a formal verification is carried out using the AVISPA tool.

Keywords:Blockchain, Internet of Things (IoT), Internet of Medical Things (IoMT), Security, Authentication and Key Agreement (AKA).

TABLE OF CONTENTS

List of Figures

List of Tables

List of Abbreviations and Symbols

Abbreviations

IoT	*Internet of Things*
IoMT	*Internet of Medical Things*
AKA	*Authentication and Key Agreement*
EPS-AKA	*Evolved Packet System based Authentication and Key Agreement*
EAP-AKA	*Extensible Authentication Protocol based Authentication and Key Agreement*
MEPS-AKA	*Modified Evolved Packet System Authentication and Key Agreement*
MITM	*Man-In-The-Middle*
AVISPA	*Automated Validation of Internet Security Protocols and Applications*
IoV	*Internet of Vehicles*
RPCA	*Ripple Protocol Consensus Algorithm*
UNL	*Unique Node List*
BFT	*Byzantine Fault Tolerant*
IMSI	*International Mobile Subscriber Identity-Catcher*
ESL	*Ephemeral Secret Leakage*
HECC	*Hyper Elliptic Curves Cryptography*
ECC	*Elliptic Curves Cryptography*
ECDSA	*Elliptic Curve Digital Signature Algorithm*
PoW	*Proof of Work*
RFID	*Radio Frequency Identification*
CK	*Canetti-Krawczyk*
XOR	*Exclusive OR*
DoS	*Denial of Service*
PoS	*Proof of Stake*
DPoS	*Delegated Proof of Stake*
KDC	*Key Distribution Center*
TA	*Trusted Authority*
GN	*General Node*
CS	*Cloud Server*
3GPP	*3rd Generation Partnership Project*
DY	*Dolev Yao*
TPD	*Tamper-Proof Device*
A	*Adversary*
HEC-DSA	*Hyper Elliptic Curve Digital Signature Algorithm*

UDP	*User Datagram Protocol*
VM	*Virtual machine*
SCC	*Supply Chain Council*
SCM	*Supply Chain Management*
P2P	*Peer-to-Peer*
SCP	*Stellar Consensus Protocol*
PoI	*Proof of Importance*
PoET	*Proof of Elapsed Time*
TEE	*Trusted Execution Environment*
CFT	*Crash Fault tolerant*
UE	*User Equipment*
DN	*Data Networks*
AMF	*Access and Mobility Management Function*
SMF	*Session Management Function*
UPF	*User Plane Function*
AUSF	*Authentication Server Function*
PCF	*Policy Control Functionality*
AF	*Application Functionality*
SMF	*Session Management Functionality*
UDM	*Unified Data Management*
SQN	*Sequence Number*
HN	*Home Network*
SN	*Serving Network*
IMSI	*International Mobile Subscriber Identity*
USIM	*Universal Subscriber Identity Module*
CL-AtSe	*Constraint-Logic-based Attack Searcher*

I INTRODUCTION

I INITIAL CONSIDERATIONS

With the emerging worldwide use of Internet of Things (IoT) scenarios in many industries such as vehicular and energy ones and for the sake of this research, medical networks in the form of Internet Of Medical Things (IoMT), with the aim of achieving smarter and more efficient systems and applications. However, concerns over security in communication between the IoT devices have been been raised due to limitations on hardware capacities of the entities that compose the network, security issues in wireless communication in heterogeneous networks, and increasing cyber security threats. Moreover, the fast increase in wireless scenarios (e.g., emergence of 5G technologies) has demanded a safe, efficient, and secure non-wired communication. Blockchain has been extensively discussed due to its start the booming related to its usage in cryptocurrency, successful adaptability to new industries, and development of information sharing technology - according to several authors, it can be a secure solution to many communication scenarios.

Blockchain can be defined as a distributed ledger that maintains permanent records of transactions executed and processed in a network and, from our perspective, a viable solution to the current vulnerabilities of IoMT. It has also enabled our proposal of an efficient authentication protocol, since the literature lacks studies on security mechanisms adapted for such scenarios and deep research on blockchain techniques. Blockchain efficiently delivers security towards vulnerabilities and enables the implementation of more advanced techniques such as signcryption, Hyper Elliptic Curve Cryptography, session-based security, auditioning of data, and all transactions of relevant information in the network. It can be applied to an IoMT network, in a 5G scenario, as in [30], thus showing its capacity for adapting to technological standards in wireless communication, used to support the aforementioned network transactions.

II MOTIVATION

This reseach hass been motivated by the lack of authentication protocols designed and adapted for IoMT wireless communication with blockchain. Traditional schemes such as EPS-AKA support neither the increasing demands of the scenario, nor the emerging cyber security threats. When working with 5G for the wireless architecture, 3GPP proposed the authentication procedure using EAP-AKA' and 5G AKA, which are protocols are based on shared key cryptography. In [9], the 5G AKA protocol has been found with authentication problems due to the lack of integrity protection for service network identities. [13], reported the 5G AKA mechanism of the sequence number (SQN) can be exploited (explored) with some specific replay attack due to its Exclusive-OR and lack of randomness. According to [9], ithe protocol guarantees privacy and security properties, although it can be improved through reductions in handover costs, delay, and energy consumption. Internet of Things (IoT) aims to offer several new applications to industries, such as healthcare, energy distribution, manufacturing and vehicular. The security of the devices belonging to these industries must be assured towards the success of IoT in e-health, since governments and private

institutions have attempted to address threats of cyber security faced by such technology and adoption of international standards of information security. Blockchain can provide such international standards, new features, and practices to solving emerging threats, as well as an efficient delivery of service. The establishment of a proper network among devices requires an adequate authentication, since not all devices are trustworthy, which might cause loss of data and security flaws. The assurance of trust demands the development of a new and appropriate authentication protocol that enables a modifications in architecture, security, and performance for IoMT and its many applications.

III OBJECTIVES

The general objective of this research is to propose a new authentication protocol for IoMT communication with Blockchain technology considering different scenarios for its implementation. It introduces a generic model in which devices can be organized into groups towards a better authentication aided by fog servers, fulfilment of IoMT security requirements, and good performance in comparison to other existing protocols.

III.1 Specific Objectives

The specific objectives are:

Generation of a new authentication protocol for IoMT communication to be used in different scenarios;

Application of security concepts towards the achievement of confidentiality, integrity, privacy, protection to several attacks (e.g., replay and identity-based ones), and other security and functionality features;

Proper implementation of blockchain in the proposed protocol, taking into consideration its functionality and the steps required for its construction;

Evaluation and comparison of available protocols regarding general characteristics, security, functionality, and computational, communication, and storage costs; and

Validation of the protocol by AVISPA for a semiformal verification.

IV CONTRIBUTIONS

The main contributions of the research involve:

1. Discussion of authentication protocols in IoT and IoMT communications;
2. Discussion of the use of blockchain for authentication procedures;

3. Proposal of an authentication protocol for IoMT with use of blockchain, fog and cloud servers, and group authentication towards improving security and performance in the communication of such scenarios;

4. Evaluation of the protocol regarding security, functionality, and computational, communication, and storage costs; and

5. Its informal and semi-formal validation.

V ORGANIZATION

The remainder of the paper is organized as follows:

. Chapter 2 presents the theoretical background addressing relevant concepts of authentication, security, and other features for the development of our scheme.

. Chapter 3 introduces a Blockchain Enabled Authentication Protocol for IoMT (BEAP-IoMT) that aims at increasing security and functionality features and improving performance in comparison to other protocols. It considers a generic network model by using fog and cloud servers, blockchain, group and batch authentication, and direct access to the 3GPP network for 5G communication.

. Finally, Chapter 4 provides the conclusions of the dissertation.

II THEORETICAL BACKGROUND

Abstract

This chapter discusses important concepts such as authentication, encryption, blockchain, security objectives and attacks, communication, network model, threat models, 5G, cryptographic techniques, and additional technologies for the understanding of the proposed scheme.

I SECURITY OBJECTIVES

The following three fundamental security objectives must be achieved for the management of a secure system[48], these are:

1. Confidentiality - assurance that information is accessible and available to only authorized members. Information must be verified and protected from attackers and other ill-intentioned entities for avoiding leakage of secret data and data manipulation.
2. Integrity - assurance that information has not been manipulated, modified, or destroyed by non-authorized entities, its source is authentic, and the origin of data is non-repudiated.
3. Availability - assurance that the system is operating in accordance with expected predictions and by authorized entities, whenever their use is needed.

Below are some other security objectives that complement the aforementioned pillars for secure environments:

- Non-repudiation: assurance that an entity cannot deny the origin of determined message or information.
- Privacy: assurance that the information of an entity is protected from unauthorized individuals.
- Anonymity: assurance that the real identities of individuals involved in a system are not disclosed and, therefore, impersonated by ill-intentioned individuals.
- Trust: asssurance that an entity or individual can fulfill commitments related to security, functionality, and communication.
- Backward and Forward Secrecy: assurance that information is secure in previous and subsequent sessions through the use of secret keys in each authentication session. Therefore, even if a current key has been disclosed, the information exchanged in previous or future sessions cannot be accessed, since the validity of each key expires at the end of each session. We consider Perfect Forward Secrecy and Perfect Backward Secrecy (PFS/PBS), when an intruder, if able to gain access to the data of a communication session between two or more entities, is unable to affect past and future sessions, respectively, with said data.

II BLOCKCHAIN

The emerging blockchain technology has been extensively researched and documented, due to its delivery of a suitable solution to authentication and access control for distributed environmentss such as IoT [42] or IoMT [20][3],its decentralized nature, cryptographic properties, and immutability. Although the idea of a decentralized chain of cryptographic nodes dates back to 1991 [29], blockchain has been implemented as the backbone of cryptocurrency via Bitcoin. However, experts worldwide see its potential for fortifying privacy and security issues in several scenarios beacuse of its improved reliability, unforgeability, fault tolerance, distributed implementation, and accessibility. Its implementation with fog computing is considered one of the most stable security solutions for time- sensitive scenarios, as in IoV[7][27], and its with smart contracts offers a more detailed and controlled access control over distributed objects. One of the main issues of blockchain technology is to make use of all its resource and processing costly features, as an energy and computational efficient solution, hence the desire for working alongside fog and Edge computing environments, as a solution to these issues. [45].

II.1 Blockchain Transactions

Blockchain holds the interactions of two or more parties in its chain of records. Such interactions, called transactions, are always performed and validated by the blockchain network and the results are added to the chain, obeying an atomic structure; in this sense, either the consensus mechanism used by the chain validates (or not) the full transaction; moreover, transactions are inspectable and independent of each other, and each method requires an address to the caller, since all data are permanent and immutable.

II.2 Blockchain Architecture

Blockchain can be seen as record storage environment, composed with blocks of data of chunks of information registered with valid timestamps. The blocks are identified by a unique hash - every block references the hash of the previous block, thus forming a cryptographic chain of blocks. A block in the blockchain will be usually composed by a block header, a hash value of the previous block header, the root of the Merkle hash tree, and the block payload, whose content depends on the consensus protocol and the objectives of the blockchain, but consists mostly of the transactional data and its requirements. Figure 1. illustrates the most commom structure of a blockchain. Users must use their private/publickeys to interact

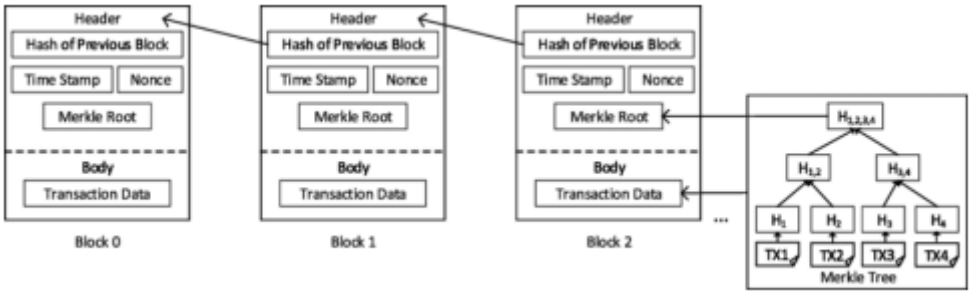

Figure 1: Blockchain structure diagram according to [41]

with the blockchain towards transaction signing and identification. The transactions are then broadcasted to the neighboring peers, where they will be validated according to the application. They are then either discarded, if not approved, or packaged into a candidate block. The process, commonly called “mining”, is shaped by the consensus strategy of the network, which decides if the candidate block is a valid one through communication with the entire blockchain. If successful, the new block is added to the chain.

II.3 Merkle Tree

A binary hash tree, also known as a Merkle tree, is a tree in which each leaf node is labeled with the cryptographic hash of a data block and each non-leaf node is labeled with the hash of its child node’s labels. It is used in blockchains for encrypting data more efficiently - each block knows the hash of the previous block while still being secure. In the blockchain context, it is a data structure that contains the hash of various data blocks that summarize all transactions in a block and enables a quick and secure validation of data and their consistency. A Merkle tree totals all transactions in a block and generates a digital fingerprint of the entire set of operations so that the user can verify whether it includes a transaction in the block. They are built from the bottom using Transaction IDs, which are the hashes of individual transactions - every non-leaf node is a hash of its previous hash and every leaf node is a hash of transaction data. From the bottom to upwards, the child node is the hash of the pairing of the hash of the two previous nodes. The process repeats until the very last node at the top, which will be the Merkle tree root and represents all transaction data of the block.

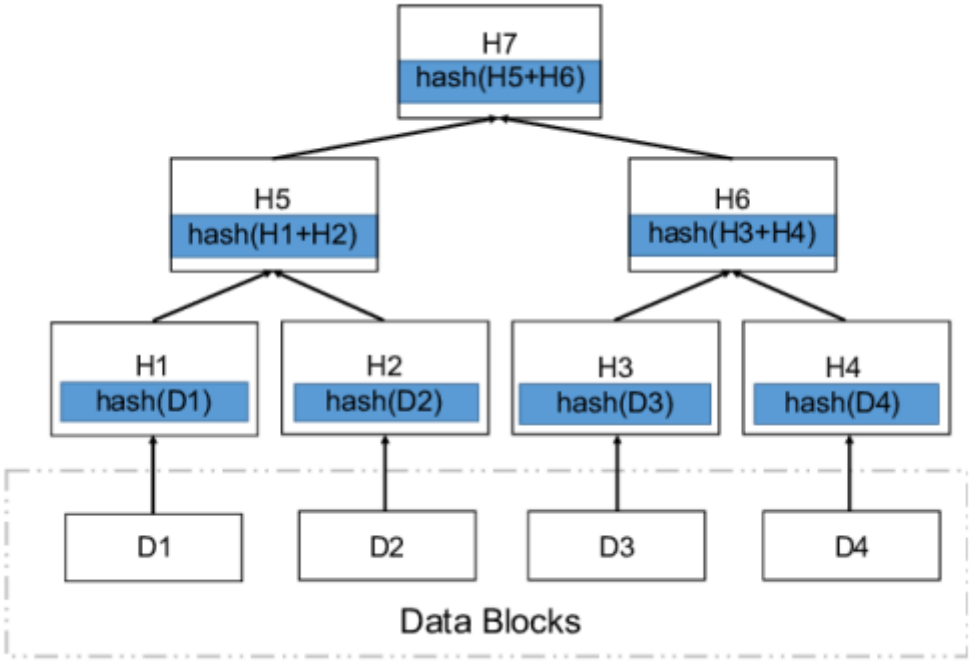

Figure 2: Merkle Tree structure according to [32]

The main benefits of a Merkle tree in blockchain are an efficient and secure validation of the data and their integrity, a reduced use of disk space, and reduced costs of communications in the network [10]. An

illustration of the Merkle Tree can be seen in Figure 2.

II.4 Blockchain classification

While Blockchains are subject to constant changes due to consistent research, some categories and nuances have already been established by academics. This section discusses such a topic and presents an overview in Tables 1 and 2.

II.4.1 Permission-based Blockchain

Blockchain can be permission-based, according to a classification which determines how the participants can interact with the chain, and its restriction procedures. In this sense, they can be either:

1. Permissioned Blockchain:for a proper participation in the network, permissioned chains require authorization of other participants, i.e., the blockchain requires a consensus and governance body with some degree of trust of authorized nodes, which makes this type ideal for internal business or organizations and private and consortium-based blockchains, facilitating the application of more effective consensus algorithms.

2. Permissionless Blockchain: any node is allowed to participate in the network without prior authorization, usually requiring a more demanding consensus mechanism (e.g., PoW) to operate, since the objective is to work in a public governance setting and in a zero-trust environment.

Table 1: Comparison of permission-based Blockchains according to [11]

Distinguishing features	*Permissionless blockchain*	*Permissioned Blockchain*
Participation	Facilitates free join and exit	Authorized participation
Transparency	Open and exit	Open/Closed
Consensus Techniques	PoW, PoS	BFT
Number of Readers	High	High
Number of Writers	High	Low
Number of Writers	High	Low
Number of Untrusted Writers	High	Low
Network Size	Huge (greater than thousands)	Limited (tens to hundreds)
Network Synchronization	Asynchronous / Partially synchronous	Partially / Fully synchronous
Network Connectivity	Loosely connected	Fully connected
Transaction Capacity	Low	High
Throughput	Low	High

II.4.2 **Participation-based Blockchain**

Blockchain can be classified according to the participation scheme of its members, of which the most popular is public blockchain, since it enables a large number of nodes with large decentralization. The main classifications are:

1. Public Blockchain: a public chain allows a large number of participants who engage in a consensus mechanism by reading and applying valid transactions. Popular examples are Bitcoin and Ethereum cryptocurrencies;

2. Private Blockchain: a private blockchain has a ruling entity with execution and write permissions, while a read authority is kept public to all members. As such, it displays a more centralized structure that resembles other distributed databases and is common in corporate and closed organizations managed by a trusted party;

3. Hybrid Blockchain: as suggested by the name, a hybrid chain has qualities from both public and private blockchains and uses permission and permissionless-based systems. A segment of the chain is controlled by an authoritative entity, whereas other segments are public. Many times, this might require the creation of a main chain, and a side chain, where each one shall take the participation rule in consideration, and have some type of communication between the chains must be provided. Although differently from public and private blockchains the solution can provide desirable results, it is not highly efficient and the setting of an infrastructure may be more costly.

4. Consortium Blockchain: A consortium Blockchain, or semi-private blockchain, is a more distributed approach to private chains for larger organizations. This federated blockchain is a type of hybrid blockchain managed by more than one entity. It has the advantages of a private structure, with increased scalability; however, it provides the network with less flexibility and more vulnerabilities.

Table 2: Comparison of participation-based Blockchain according to [11]

Distinguishing features	*Public Blockchain*	*Consortium Blockchain*	*Private Blockchain*
Infrastructure	Highly-decentralized	Decentralized	Distributed
Governance type	-	Public consensus managed by a set of participants	Single owner manages the consensus
Security	Proof of Stack	Proof of Work	Pre-approved participants
Asset	Native	Native	Any
Participation in consensus process	No authentication	Authentication required	-
Throughput	Low	High	High
Consensus Algorithm	Without permission	With permissions	¨With permissions
Identity	Pseudo-anonymous	Approved participants	Approved participants
Data Immutability	Possible, no rollback	Possible, has rollback	Possible, has rollback
Network Scalability	High	Medium to low	Medium to low
Transaction Processing Speed	Slow	Fast	Fast
Access	Public read/write	Restricted	Restricted

II.5 Consensus algorithms

Due to the nature of distributed systems, the nodes must agree on transactional and execution data, and, therefore, a consensus must be reached for all components of the system. This is valid for blockchains, since they rely on a consensus protocol for ensuring all nodes in the network agree on the transactional records, given the influence of desynchronized, malfunctioning, and malicious nodes. In other words, the agreement of the chain regarding the order of transactions is not enough, for they must also account for faulty elements in the procedure [10]. A Byzantine Fault Tolerance (BFT) protocol follows the concept of protocols for computer science based on the Byzantine Generals Problem [39], in which n Byzantine generals are preparing to attack a fort, and they must agree upon the same course of action, otherwise it would lead to a failed attempt; it also must take into account that the generals are far apart, with unreliable communication and there could be malicious generals in their midst. Therefore, a BFT protocol acts as a method for consenting not only on the agreement of data, but also on malicious and faulty nodes, called byzantine errors [39]. A myriad of consensus protocols has been developed for blockchain scenarios, which is essential for a secure and efficient implementation of the chain in any environment. In what follows is a discussion on some of the main and most formalized protocols.

II.5.1 Proof of Work (PoW)

Introduced with Bitcoin, PoW works through the solution of many complex calculations by the nodes. Such calculations are usually a challenge offered by the blockchain towards the validation of its nodes and transactions – the more complex, the more secure. In Bitcoin, the system scans for a value that, when hashed, has a hash starting with a number of N leading zeros, where N determines the difficulty of the calculations. This is accomplished through the addition of a nonce to the original value until the resultant hash starts with the requisite number of zero bits. Once it has been done, the block cannot be changed without redoing the work for that specific block and all blocks that come after it. There is an initial "genesis block", whose hash consists entirely of zeroes, and all further blocks have a hash that consists of the previous block's hash alongside the nonce required for the calculation. In PoW, the chain with most work is the main chain [6][52]. Whereas Proof of Work is highly decentralized, stable, and secure and theoretically requires the attacker must have at least 51% of the total computational power of the network, such a feature is highly demanding for the network in both computational overhead and energy consumption. Its implementation is infeasible in long term, especially for big chains [46][6][52]. Furthermore, recent research has shown PoW chain cannot be overtaken with 25% of the computing power of the system [6].

II.5.2 Proof of Stake (PoS)

Blockchain with its consensus process can adopt the leader election mechanism to reduce the amount of mining necessary in the system. The design of PoS is hybrid design, following the idea of PoW, but using currency age to determine the hashing difficulty of mining. The more assets of the blockchain owned, the longer the holding time for transaction submissions, which means easier mining of the block. To be effective, PoS chains require a large number of blocks – the larger the number, the more organic the

accounting. As the currency is spent, its age is reset to zero and the validator can pay for the privilege of holding assets of the blockchain, thus having a larger stake in the system. Unlike PoW, the main chain of PoS is the one with highest consumed age. The hashing challenge of PoS is calculated by Eq.1.

$$\textit{proof of hash} < currency \times age \times target \tag{II.1}$$

PoS can work as an interest-based currency and, therefore, saves resources, since the mining does not waste so much energy. It is efficient and scalable and requires only the equity proof, which greatly reduces the time for consensus confirmation and can organically increment new blocks. However, the implementation of PoS is more complex and may lead to more security breaches. It also demands incentive to properly work with its features. In PoS, hypothetically, an attacker would require the control of 51% of the stakes of the network to take over the chain. [6][52] Although Delegated Proof of Stake (DPoS) is similar to PoS, the leader node can elect several agents to help with the consensus procedure, highly improving security. However, it reduces its distributed nature, creating points of failure, and also opens to bribery issues of its agents, which, alongside the reliance on tokens, forces the DPOS system to be implemented in a specific and highly monitored scenario.[52] Also, of the interest of this paper, many issues have been found in the implementation of PoS for blockchain in IoT scenarios, as there is still not a reliable way to implement a system of stakes on IoT settings. [46].

II.5.3 Practical Byzantine Fault Tolerance (PBFT)

According to the previously discussed BFT scenario, PBFT is an algorithm based on service as a state machine, where the different nodes of the distributed system conduct the replication of the state machine, having R total replicas in the environment. Following the BFT condition, R = 3f + 1, where f is the maximum number of faulty nodes, meaning the system with R nodes can tolerate f Byzantine errors. PBFT blockchain relies on a primary master node responsible for receiving requests towards multicasting its requests to the other nodes, known as secondary nodes, which reply if successful or not. If the client responsible for the initial request receives f+1 replies with a same answer, it validates those data and receives it. In a blockchain, the client may be the block wishing to join the chain. This is a classical algorithm, in which the state machine analysis leads to high and consistent performance and a secure system in comparison to several other consensus algorithms. However, it requires previous knowledge of the number of nodes to execute for connecting them, which leads to poor scalability, since it cannot handle dynamic nodes well, making it very ineffective to implement in public Blockchains [52]. The need for master nodes also reduces the decentralization of the chain, which must be closed for the running of the algorithm.

II.5.4 RPCA and XRP Ledger Consensus Protocols

XRP Ledger Consensus Protocol is formerly known as the Ripple Protocol Consensus Algorithm (RPCA) and was designed for use in Ripple cryptocurrency, for addressing latency issues present in other algorithms [6]. It functions on a ledger basis, where the acting server nodes in the running algorithm are set on an Unique Node List (UNL).

The XRP ledger consensus protocol, uses many of the RPCA mechanisms, while improving them on

some technical aspects. It operates with a open ledger, that has the current set of transactions on the nodes that still require validation, and also the last-closed ledger is used, as it has the most recent ratified set of the consensus, representing the current state of the network. The protocol works in rounds, initially verifying all currently valid but not applied transactions, and makes them public in a list, called Candidate Set. Every server must gather the candidate set of all servers on the network and then vote on the veracity of the transaction – those that score a pre-defined minimum percentage of the votes as "Yes" go to the next round; otherwise, they are either discarded, or included in a future candidate set for a future election. In the final round, a minimum of 80% of positive votes must be achieved, so that the transaction can be finally absorbed on the blockchain. However, XRP offers an improvement over RPCA mainly in electrical usage, transaction cost and scalability issues found in the latter, and refinement in several procedures [17].

Another issue related to RPCA refers to the achievement of a trustworthy voting process. A minimum 20% overlapping of UNLs used to be required; however, in recent years, it has been disproved, due to vulnerabilities and other flaws found on the algorithm. XRP offers a revision upon these issues, claiming an at least 90% overlapping of the UNL for guaranteeing safety [17]. XRP is a BFT algorithm; therefore, it takes into account fault tolerance for defective and malicious nodes [17].

II.5.5 Other Consensus Protocols

Other consensus protocols have been proposed and adopted, such as Stellar Consensus Protocol, Proof of Importance and Proof of Elapsed Time. Among such protocols, there are also hybrid consensus protocols, such as Aethernity and Aelf.

Stellar Consensus Protocol (SCP) is a consensus protocol whose nodes choose which nodes to trust - the group of nodes with trust is a quorum slice and quorum is the set of nodes sufficient for reaching an agreement.

Proof of Importance (PoI) has been used for many cryptocurrencies , such as XE. The consensus mechanism requires that each block of the network must have an token value that can be vested or unvested, by other entities. A fraction of the unvested total of all accounts is transferred to vested at an assigned interval of blocks. A calculation of the importance value of the blocks will be calculated according to the rank of the account, a weight factor, and constants [6].

Proof of Elapsed Time (PoET), developed by Intel and used in platforms such as Hyperledger Sawtooth, is a consensus algorithm whose proof of work is replaced by a randomly generated wait time via a Trusted Execution Environment (TEE).

Raft is a distributed consensus algorithm that had in its core, to be designed to be easily understood, and as such, aims to solves problems by splitting the workload in minor processing tasks, performing a simple level of load balancing. Developed on the basis of Paxos consensus towards obtaining multiple servers for agreeing on shared states in the face of failures, it shows very high performance and is a leader election system for log replication. While Raft is fault tolerant, it is not BFT, rather its Crash Fault tolerant (CFT) tolerating up to (N - 1)/2 crashes in the network nodes, however, unlike BFT, it does not consider malicious criteria on nodes.Raft is also tax-reliant and can be used in systems in which currency is circulated; therefore, it is not ideal for some blockchain scenarios.[33]

An important idea bet on by several researches regards hybrid consensus, according to which multiple consensus are used for different steps of the blockchain design. Examples include Aethernity and Aelf consensus protocols that use PoW and PoS together – the former uses PoW to generate blocks and PoS for major decisions of the chain, whereas the latter employs a PoS chain for management and PoW for the raw execution power.

III P2P NETWORK

Blockchain adopts a decentralized, load balanced, fault-tolerant P2P network that resembles a small-world model with average feature path length and large aggregation coefficient [11]. The network nodes can be divided according to their authority and function in the chain (e.g., write, read, execute, among other capabilities). The architecture ensures a dynamic and stable network, which can preserve the transaction data integrity, while assuring a decentralized solution. However, the P2P architecture shows some vulnerabilities, such as permission for malicious nodes to interact with the network, thus damaging it, via collusion with other participants, or known attacks. It also suffers from reliance of a distributed infrastructure, which, without proper deployment, may result in a waste of resource, since blockchain demands a relatively large data overhead, computer and communication costs [11][38], and high energy consumption [6]. However, [35][36] claimed it can be substantially improved with a proper use of consensus, network design, and deployment.claimed it can be substantially improved with a proper use of consensus, network design, and deployment.

IV 5G NETWORKS

5G is the fifth generation of cellular wireless technology and aims to offer a massive boost to throughput and integrity of connection. Its integration can bring real-time data transfer to existing infrastructures and, therefore, is vital to time-sensitive solutions such as IoV and IoMT [54]. The 5G architecture is composed of User Equipment (UE) (e.g., smartphones and other devices), which connects the 5G Radio Access Network to the 5G core and then accesses further Data Networks (DN). The UE has an entry point in the Access and Mobility Management Function (AMF) that selects the Session Management Function (SMF) for managing the user session according to the service currently used. The User Plane Function (UPF) transports traffic between the UE and the outside network and the Authentication Server Function (AUSF) handles the authentication of UE and access services to the 5G core. Other functions handle Policy Control Functionality (PCF), Application Functionality (AF), Session Management Functionality (SMF), and Unified Data Management (UDM). An illustration of the 5G architecture can be seen in Figure 3 below.

IV.1 5G Authentication

The 3rd Generation Partnership Project (3GPP) standardizes 3G, 5G, and 5G cellular technologies brought forth the Authentication and Key Agreement (AKA) protocol, used to provide mutual authenti-

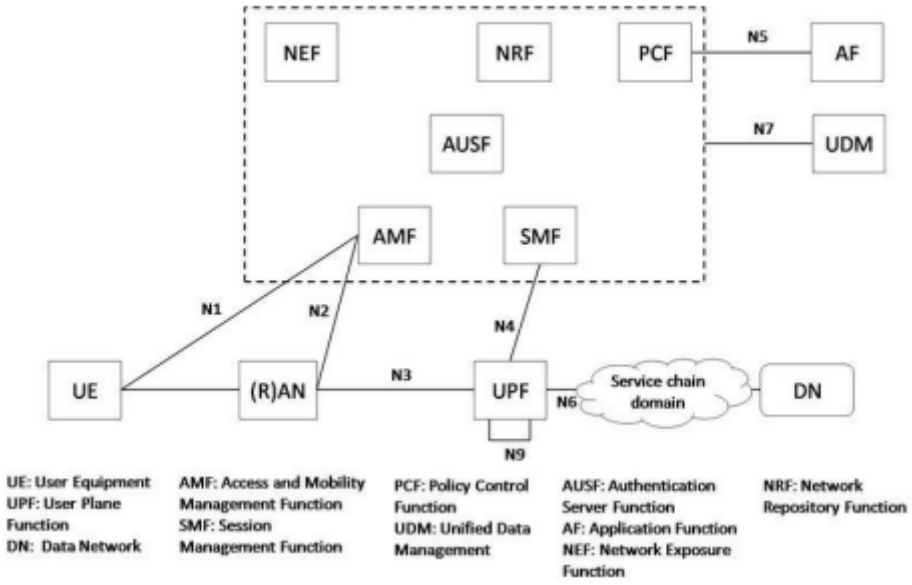

Figure 3: 5G architecture diagram, as seen in [14]

cation of devices and key generation for future communication. Together with AKA, the Extensible Authentication Protocol (EAP) is used for a secure P2P authentication as EAP-AKA. Evolved Packet System Authentication and Key Agreement (EPS-AKA) and Modified Evolved Packet System Authentication and Key Agreement (MEPS-AKA) protocols are evolutions of the previous mechanisms and 4G technologies. AKA is a mostly symmetric cryptographic and Sequence Number (SQN)-based protocol. Vulnerabilities of those techniques have been detected along the years [12], thus requiring improvements in AKA methodology by 3GPP. 5G AKA protocol introduces randomized asymmetric encryption as a solution to previous weaknesses. Figure 4 illustrates the AKA protocol architecture, where the diagram consists of UEs representing the 5G service subscribers, the users of the networks, i.e., Home Networks (HNs), are the storage of the subscriber identities, and their USIM card denotes the local base station of the cell. In case of no base station, the Serving Network (SN) is issued, allowing UEs to attach to and play the role of relays to HNs. Every USIM card contains encrypted identification values, such as MAC, International Mobile Subscriber Identity (IMSI), corresponding SQN, and symmetric key (K_{IMSI}).

Borga [12] and Basin [9] reported the existence of several vulnerabilities in AKA protocol (e.g., unprotected identity requests (IMSI-catchers), linkability of failure messages, weaknesses in the agreement properties between UEs and SNs, device exposition due to the implicit authentication mechanism, and attacks as presented in [9] shows that there are privacy issues to be adequately treated. [54] discussed the way those issues lead to 5G vulnerability taking into account several attacks (e.g., Man-In-The-Middle, identity, privileged insider, malware, session-based ones, physical device stealing, among others) which, together with other questions, make 5G unreliable on networks with not much endpoint and session security, such as IoT [54]. According to Wazid [54], Guo [45], Kang[37], Hammed[30], and Rahman [47], blockchain can work as a solution to the security issues found on 5G networks, since its characteristics (discussed elsewhere in this paper) provides privacy, endpoint and session security, and protection to several attacks, such as MIM. Rahman et al. [47] and Hammed et al. Hameed [30] proposed frameworks for a secure integration of 5G and blockchain in a distributed ledger and the integration of 5G blockchain with e-health

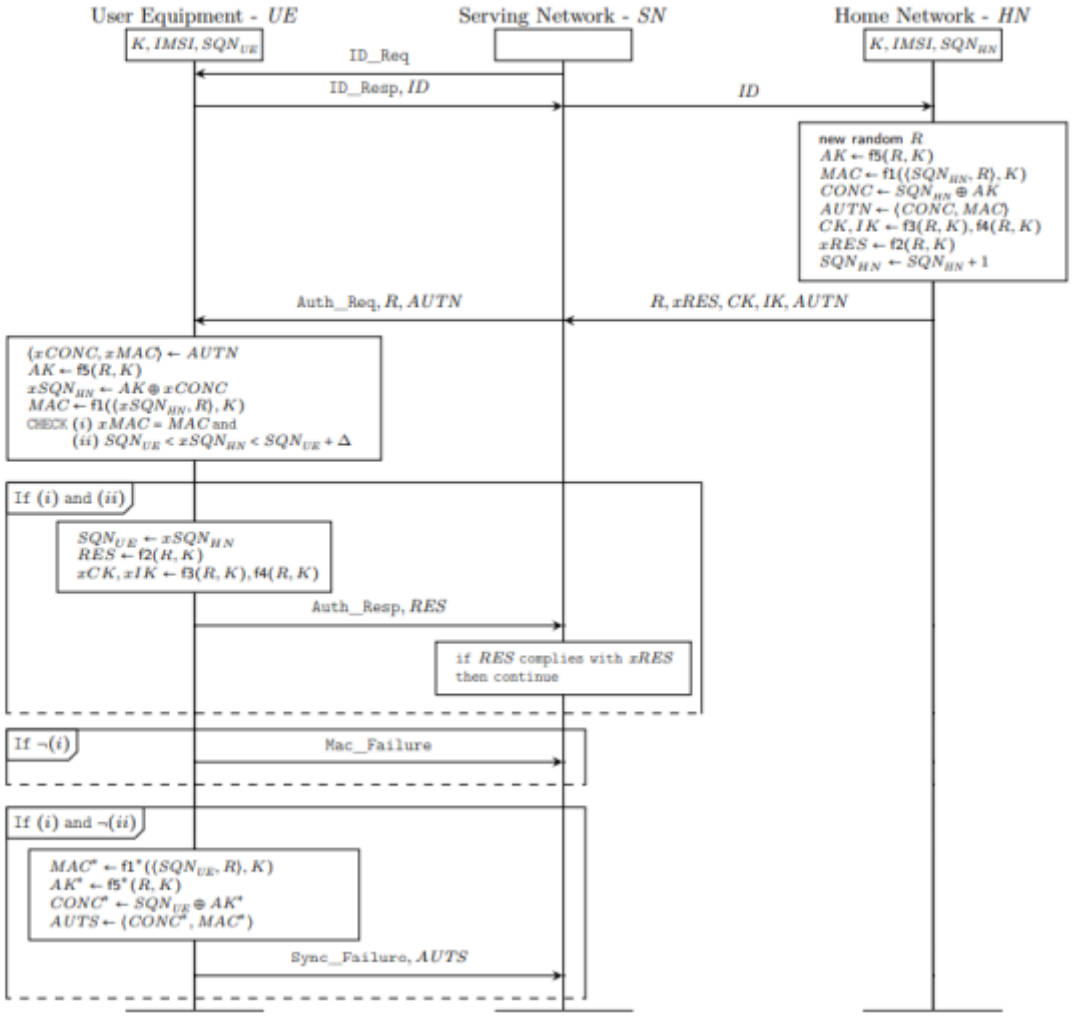

Figure 4: AKA protocol architecture, as seen in [9]

- specifically, IoMT and telemedicine.

V FOG NETWORKING

Due to the increasing demand for fast responses to decentralized devices, the use of edge devices has been the focus of several discussions and a fundamental part of edge computing. Fog networking, or fog computing, is an extension of edge concepts that brings a cloud architecture closer to on-demand locations and provides a much higher quality of service mainly regarding reduction in latency and increase in processing power and redundancy in case of failure. Its use for a fast response, high power processing solution, is a viable option for the execution power necessary for the application of blockchain consensus mechanisms in IoT scenarios, which consist mostly of weaker devices in computational power [45].

VI THREAT MODELS

Discussions on security protocols require the definition of threat vectors and vulnerability surface and other considerations for the setting of the paper plans. Dolev-Yao (DY) model [21], the most widely addressed threat model related to security protocols, especially in the communication and authentication spectrum, sets two communicating parties that speak with each other via an insecure channel. End point devices and users are usually not trustworthy for communication. The model also sets an adversary (A) that eavesdrops, modifies, and deletes messages in that channel. Canetti-Krawczyk (CK) [15],a more robust model, is considered the de-facto model regarding authentication via key agreement [14]. According to it, A has all capabilities of DY model, can potentially compromise credentials and session states of the communication channels, and physically captures endpoint devices via power analysis attacks, which enables the occurrence of other attacks. Among the several attacks to be considered in heterogeneous and distributed communications are identity-based ones such as impersonation, in which an adversary masquerades as an authentic entity in the system, and replay, in which packets previously sent are intercepted and relayed by the intruder via spoofing of the identities of the devices.

Eavesdropping attacks can occur via sniffing of data packets, thus enabling future ones (e.g., ID and key theft), and Denial of Service (DOS) is performed via either flooding of the server with many requests, or updating the server with incorrect information for taking down the service availability, which is highly dangerous for time-sensitive services, like IoMT. Despite being a solution to several attacks, Blockchain shows vulnerabilities such as Sybil attacks, in which an adversary subverts the system reputation by creating many identities within the system and gaining influence over the decision-making of the setting, Eclipse attacks, according to which an attacker controls a large number of IP addresses and creates partitions within the blockchain network, singling out miners from the remainder of the chain, and mining attacks (e.g., mining pool and selfish mining), in which miners that are part of the blockchain can use their resources to exploit vulnerabilities in the network [11].

VII CRYPTOGRAPHIC TECHNIQUES

Due to increasing necessity of faster and more secure communications for larger networks with lower computational resources, new cryptographic solutions have been designed for improving authentication. The following technologies were largely used by Pardeshi [45], Khalid [38], Kang[37], Garg [25], and Bagga [7]:

1. Bilinear Pairing, which provides stronger security mechanisms to devices with low computational power, focusing on efficiency and security. It can be used for impersonation attacks, MitM, and other identity theft techniques.

2. Elliptic Curve Cryptosystem (ECC), which offers high security for key exchange authentication solutions using much smaller sized keys.

3. Hyper Elliptic Curve Cryptosystem (HECC), which is an extension of ECC and relies on the breakage of the discrete logarithmic problem in the jacobian curve, providing a high degree of complexity.

However, instead of forming a group of points for solving the polynomial problem, it develops jacobian groups, which include a divisor as the generator and a point leading to infinity. Therefore, group operations in HECC are divisor-additive and divisor-dubling, instead of simply additive and multiplicative operations of ECC. HECC are superior due to their smaller key size, but still provides equal level security [19].

4. Signcryption, which is a primitive cryptographic function that provides confidentiality and authentication for messages over an unsecure channel by performing signature and encryption in a single step, usually employing cryptosystems of small key sizes such as ECC and HECC. It can reduce the number of steps and computational power necessary for authentication operations, thus being useful for resource-constrained systems, but still providing public verifiability, forward secrecy, confidentiality, non-repudiation, integrity, and other security capabilities [19].

5. Hashing, which ensures privacy, security, and integrity to data transmitted on the network. One-way functions should be used to ensure forward secrecy, since it aids in masking the original input.

VIII CONCLUSIONS

According to the overview of the theoretical background of this study, a proper implementation of an authentication protocol requires the achievement of some security objectives such as confidentiality, integrity, and availability. Blockchain is an advanced, but still relatively new technique, and, therefore, enables several options for its construction. It has been the object of many studies, for it offers security, performance, and integrity to complex network settings. The consensus mechanisms of blockchain have been a topic of debate, and the original use of PoW mechanisms is not viable in most scenarios, as it is very demanding in terms of costs, and does not offer long term sustainability. The benefits of 5G networks include fast and reliable communication; however, it faces security issues on its current 3GPP implementation. Blockchain has been considered a solution towards stabilizing 5G networks in more complex cases. The technologies and techniques presented in this chapter can be used to achieve new, safe, and efficient solutions to existing problems in communication, specially in 5G, P2P, and IoT environments.

III BLOCKCHAIN-ENABLED GROUP AUTHENTICATION PROTOCOL FOR INTERNET OF MEDICAL THINGS OVER A 5G NETWORK

Abstract

With the emerging paradigm of heterogeneous networks such as 5G and the rising demands for smarter, faster, and more secure technologies, shown in settings for smart cities, Internet of Things (IoT), and Internet of Medical Things (IoMT), blockchain has been a topic of research due to its ability to provide a secure, reliable, and unmodifiable solution. This work presents an overview of blockchain technologies with a focus on their integration with 5G networks and smart environments such as loMT. It also discusses the possible attacks to those settings and the solutions proposed and reviews authentication protocols from the literature, comparing their solutions, performances, and security flaws. A new Blockchain Enabled Authentication Protocol (BEAP-IoMT) validated by AVISPA tool is proposed for Internet of Medical Things over a 5G Network and analyzed regarding elements of computational, communication, and storage costs and its security and functionality features. In a comparison with those from the literature, it shows better performance of BEAP-IoMT in most cases and higher security protection.

I INTRODUCTION

I.1 Blockchain

The emerging blockchain technology has been extensively researched and documented, since it provides a suitable solution to authentication and access control for distributed environments such as IoT [42] and IoMT [20][3], due to its decentralized nature, cryptographic properties, and immutability. Experts worldwide have reported its potential for fortifying privacy and security issues in several scenarios, owing to its improved reliability, unforgeability, fault tolerance, distributed implementation, and accessibility. Its implementation with fog computing is considered one of the most stable security solutions for time-sensitive scenarios, according to IoV[7][27]. The integration of blockchain with smart contracts, and its integration with smart contracts offers a more detailed and controlled access over distributed objects.

One of the main characteristics of blockchain technology is that it brings all its features as an energy and computational efficient technological alternative, that solves needs involved in the security of the computer and communication networks domains. Moreover, three types of blockchain have commonly been considered:

. Public Blockchain: a non restrictive and permissionless blockchain, i.e., any entity can access, validate transactions, and participate in consensus mechanisms; it is used in systems such as Bitcoin and Ethereum and is the most popular type of blockchain, enabling a large number of nodes with a large decentralization;

. Private Blockchain: it is controlled by an organization or company; it is centralized, restrictive, and authoritative, and only entities predefined by the organization or company can maintain and validate the records; it is suitable for use in closed systems where all nodes (devices) trust each other;

. Consortium or hybrid blockchain: a decentralized blockchain comprised of several organizations or companies and used for semi closed systems composed of several companies such as a group of banks or government organizations.

Due to the presence of high delay on Peer-to-Peer (P2P) network, the preservation of the order of transactions becomes has become an issue.Towards solving it, the blockchain system must design a mechanism, referred to as a consensus mechanism, which agrees on the order of transactions and decides if they must be trusted in similar a time frame.

One of the cornerstones of the construction of a secure blockchain is the consensus mechanism used. Blockchain is a decentralized system and the consensus algorithm enables the many nodes that compose it to agree on the creation and joining of new blocks while also providing an incentive to its functionality. Therefore, it improves its security features, guaranteeing trust in and defining the tolerance of faulty nodes.

A myriad of consensus protocols have been developed for blockchain scenarios; some operate on the workload of the system (e.g., Proof of Work (PoW) blockchains), on the stake value of certain nodes (e.g., Proof of Stake (PoS)), and on prioritization of fault tolerance (e.g., Pratical Byzantine Fault Tolerance (PBFT) among others). However, many of those consensus mechanisms are not suited to an IoT scenario [6] [52], or medical systems, which require low latency, fault tolerance, security, and use of low processing power devices [20].

An algorithm known as Ripple Protocol Consensus Algorithm (RPCA) and developed for use in Ripple cryptocurrency, RPCA was designed for addressing latency issues present in other algorithms [6]. It functions on a ledger basis, in which the acting server nodes on the running algorithm are set on a Unique Node List (UNL), the open ledger is the current set of transactions on the nodes to be validated, and the last-closed ledger is the most recent ratified set of the consensus, representing the current state of the network. However, XRP shows an improvement over RPCA, mainly regarding electrical usage, transaction cost, and scalability issues that might be found in the latter alongside refinement on several procedures [17].

Another issue raised by RPCA was the minimum requirement of overlapping UNLs as 20%, disproved due to vulnerabilities and other flaws found. XRP offers a revision upon them and is a Byzantine Fault Tolerance (BFT) algorithm, i.e., it takes into account fault tolerance for defective and malicious nodes [17].

This study considers XRP Ledger consensus mechanism an adequate one, taking into account the aforementioned aspects and the fact it meets such requirements and operates on both private and consortium blockchains, which are more suited to IoMT networks [20].

I.2 5G

Ravishankar Borgaonkar [12] and David Basin [9], reported several vulnerabilities of AKA protocol (e.g., unprotected identity requests (IMSI- catchers), likability of failure messages, weaknesses on the agreement properties between UEs and SNs, and device exposition due to an implicit authentication mechanism) and attacks addressed in [9] demonstrated the existence of privacy issues. [54] discussed the way those issues lead to 5G vulnerability to attacks such as Man-In-The-Middle, identity, privileged insider, malware, session-based ones, physical device stealing, among others, thus making 5G unreliable on networks with not so many endpoints and session security, such as IoT [54].

According to Hammed [30], Wazid [54], Pardeshi [45], Kang [37], and Rahman [47], blockchain can work as a solution to security issues of 5G networks, since they consider its characteristics (already discussed in this paper) can provide privacy, endpoint and session security, and protection to several attacks (e.g., Man-In-The-Middle (MITM)). Rahman et al. [47] and Hammed et al. [30] proposed frameworks for a secure integration of 5G and blockchain into a distributed ledger and the use of 5G blockchain integration with e-health, specifically IoMT and telemedicine.

I.3 Proposed solution

Apart from discussing the mechanisms and current scenario of the technology, this work proposes a Blockchain Enabled Group Key Authentication Protocol for Internet of Medical Things over a 5G Network (BEAP-IoMT) with the use of 5G for a fast and reliable communication of devices of the medical network, including those in the internal hospital network and remote user equipment, so that blockchain provides the system with security, integrity, privacy, availability, and immutability. The considered medical environment is suitable for private or consortium-permissioned blockchain.

The protocol is based on the combination of three resources::

- signcryption, for encryption, decryption, signature signing, and signature verification, offering better performance to the authentication process;
- HECC (Hyper Elliptic Curve Cryptography), for increased security to the authentication process, while offering better performance than ECC (Elliptic Curve Cryptography) mostly for reductions in the communication costs of the required operations;
- blockchain, due to its decentralized and tamper-proof nature, which enables a secure network to handle IoMT communication, while offering confidentiality of data, fault tolerance, and several other security and functionality features.

The protocol offers security, better performance, and other security and functionality capabilities, while addressing the issues of an IoMT network, such as resource-constrained nature of the devices that compose it, availability, and fast response for several activities and other requirements, as presented in Hammed [30] and Dilawar [20].

It also provides protection against common attacks such as Man- In-The-Middle (MITM), identity-based ones, eavesdropping, replay, sybil ones, Ephemeral Secret Leakage (ESL), privileged insider, and

Denial of Service (DoS) and other important security features such as forward and backward secrecy, immutability for avoiding data modification attacks, integrity, availability, and privacy.

I.4 Main Contributions

The main contributions of this work include:

a) A new hybrid cryptography-based Blockchain Enabled Group Key Authentication Protocol that considers an IoMT scenario, but can act on several constrained IoT enviroments due to the use of signcryption techniques powered by Hyper Elliptic Curves Cryptography (HECC);

b) The guarantee of several security properties (e.g., anonymity, untraceability, secrecy) and resistance to attacks to both network side (e.g, MITM, Identity-based ones, and ESL) and blockchain side (e.g, Sybil attack and 51% attacks);

c) Validation of the security features by Automated Validation of Internet Security Protocols and Applications (AVISPA) and a detailed verification of other functionalities;

d) Support to a dynamic addition of identities of the network; and

e) A comparative performance analysis with other competing proposals regarding computational, communication, and storage costs and their supported security features.

The scheme aims to provide additional security with reduced or comparable computational, communication, and storage costs against other models from the literature.

I.5 Structure of the work

Section II describes some related work; Sections III and IV introduce the protocol and its blockchain construction phase, respectively; Section V addresses the dynamic capabilities of node addition to the network of the scheme; Section VI is devoted to a security analysis of the model; Section VII reports its performance analysis; finally, Section VIII provides the conclusions.

II RELATED WORK

In this section, we present an analysis of related literature. A general overview of the related work is presented on Table 3.

Khalid et al.[38] proposed a lightweight blockchain authentication protocol for IoT systems based on study cases on smart hospitals. Following a layer-based architecture, it makes use of device-to-fog, fog- to-fog and device-to-device communications. All IoT devices must be registered by the blockchain-enabled fog node for future authentication, providing access control. ECDSA generates public and private keys and

requires less power processing and storage, while providing the same level of security as Rivest-Shamir-Adleman (RSA). The authors addressed mutual authentication, authentication, non-repudiation, spoofing, sybil, message replay, and substitution attacks and aimed at the protocol´s implementation at much shorter execution time and power consumption. However, the communication costs of the network were not considered. The paper is based on the use of Ethereum blockchain‘s PoW consensus mechanism, which brought increased delay; despite having a lightweight authentication mechanism, a lightweight consensus algorithm is required.

Jangirala et al. [33] developed a scheme for secure lightweight Blockchain Enabled RFID-Based Authentication for supply chains in a Mobile Edge 5G environment using one-way cryptographic hash and bitwise rotation operations for the establishment of symmetric session key exchanges. The authors addressed Forward Secrecy, privileged internal attacks, RFID reader and tag impersonation attacks, replay attacks, MITM, and Ephemeral Secret Leakage (ESL) through a formal verification by AVISPA tool. They also adopted a raft consensus mechanism for private blockchains and reported the protocol provided added security and functionality, however, at increased communication and computational costs comparable to those of other schemes. No in-depth analysis of the effects of the consensus mechanism on the scheme was provided.

Garg et al. [25] proposed a Blockchain Enabled Authenticated Key Management Protocol for IoMT, for the management of Implantable Medical Devices (IMD) with a focus on secure sessions between the network entities.The scheme considers the CK threat model and is based on the use of non-singular elliptic curve cryptography, efficient one-way hash functions, bitwise XOR operations, and biometric security via a fuzzy extractor. Security properties such as protection against privileged insider, MitM, impersonation, ESL, physical capture and data modification attacks were verified by AVISPA; anonymity and untraceability are also provided. It achieved better performance regarding communication and computational costs compared to other protocols with the use of Ripple Protocol Consensus algorithm.

Xu et al. [55] introduced a blockchain-based authentication and dynamic group key agreement protocol, based on a generic network model for MEC networks and consisting of two parts, namely a Key Distribution Center (KDC) and groups of Generic Nodes (GNs), acting as the Supply Chain Management (SCM) node and Supply Chain Council (SCC), respectively. The authors considered a network comprised of groups, where management nodes compose the blockchain and authenticate the groups for the network. The nodes are arranged on a circular closed list, sorted by their Ids. Every node of a group sends an authentication request to their neighbor, and after all nodes have authenticated each other, they generate a group key via a key agreement of the group network. Impersonation, capture, and replay attacks were addressed and forward and backward secrecy was assured through a formal validation by ProVerif tool. Although the protocol showed improved performance and energy and resource consumption in comparison to several discussed models, it lacks a proper analysis of the blockchain regarding its consensus mechanism, or the internal application of its features. The study provides no discussion on blockchain attacks.

Table 3: Comparison among related literature

Literature	Cryptographic Solutions	Consensus Mechanism	Protection to attacks	Drawbacks / Limitations
Khalid et. al. [38]	Asymmetric Encryption with Digital Signatures; Use of Elliptic Curve Digital Signature Algorithm for key generation.	PoW, with Ethereum Blockchain	Spoofing, Sybil, message Replay, and Substitution attacks	∗ Does not provide a proper analysis of blockchain consensus mechanisms; *No mention of common attacks such as MITM, DOS, and session based ones; *Lack of a proper formal validation.; ∗ Lack of a proper formal validation validation.
Jangirala et.al. [33]	Symmetric Encryption; One-way hash; Bitwise rotation operations.	RAFT	Privileged insider, MITM, replay, impersonation, and ESL attacks.	∗No proper analysis of blockchain consensus mechanisms; ∗No BFT scenario; ∗Increased communication costs.
Garg et. al. [25]	Asymmetric Encryption; One-way hash; Bitwise XOR operations; ECC encryption.	RPCA	Privileged insider, MITM, impersonation, ESL, physical capture, and data modification attacks.	∗Use of an outdated version of ripple protocol; ∗No proper wireless network architecture specified.
Xu et. al. [55]	Hybrid Encryption; Bilinear pairing; ECC encryption.	DPoS	Impersonation, Capture, and Replay attacks	∗*No mention of common attacks such as MiTM and DoS; ∗*No mention of blockchain attacks such as Sybil and 51% attacks; ∗*No proper analysis of blockchain consensus mechanisms.

III PROPOSED PROTOCOL

Our protocol aims at a reliable authentication mechanism for IoMT networks using blockchain technology to provide the system with security, integrity, privacy, availability, and immutability. It adopts 5G for fast and reliable communications of the devices on the medical network and advanced cryptoghraphic techniques such as Hyper Elliptic Curve Cryptosystem (HECC) for adding security while reducing costs, due

to its smaller key size. A batch authentication technique based on fog servers is proposed over the grouping of several devices in a Key Distribution Center (KDC), foir providing additional computing resources, reduced latency, and overall fast response time due to time-sensitive information commonly present in the medical scenario.

In Table 4 below, we present the notations used on this work, alongside their meanings.

III.1 Network Model

The scheme's network model consists of some key elements, namely General Nodes (GN), representing devices to be grouped during their registration in the system, KDC, which encompasses multiple fog servers, blockchain, comprised of several Cloud Servers (CS), Trusted Authority (TA), and 5G core of the network, which provides the main infrastructure necessary for enabling 5G for the entire model. All GN devices are equal, have certain computing and storage capabilities, communicate in the 5G network, and join or leave a group at any time. KDCs consist of several fog servers, thus improving communication and processing workloads. The general nodes always try to communicate with the most efficient fog server regarding the distance between GN and KDC and how overloaded the fog server is, hence the servers being able to redirect the client to more adequate options that provide redundancy and more efficient workloads. TA only acts in the registration phase of the protocol and, therefore, does not influence the other steps directly. The Blockchain is maintained by a collection of CSs that act as miners, and the consensus method will be ripple, working via XRP ledger and providing a consortium-based chain towards fault redundancy and improved security and performance. The 5G network can reach all devices located in the 3GPP coverage area. An ilustration of the network architecutre can be seen in Figure 5.

III.2 Threat Model

The scheme adheres to the guidelines of the widely used Dolev-Yao (DY) model, according to which any two communicating parties interact over an open insecure channel and end-point users are, in general, not trustworthy. Canetti and Krawczyk (CK) adversary model, which considers all capabilities of an adversary (A) can be stated in the DY model and can compromise the secret credentials and states of the sessions was also followed. A can physically capture the devices that compose the general nodes section of the network and extract their stored information, thus obtaining confidential information that may compromise the network. Fog servers are assumed to be Tamper-Proof Devices (TPD), and, as such, even if A can capture KDC, they cannot extract any secret information. TA is considered a fully trusted entity; therefore, the network is not compromised. Cloud servers act as miners of the blockchain and are also considered trusted entities.

III.3 Preliminaries

- Bilinear pairing: Let G_1 and G_2 be cyclic additive and multiplicative groups of prime order q respectively. The generator of G_1 is g_1. Let e: be a bilinear pairing, which satisfies the following properties:

Table 4: Notations and their meanings.

Symbol	Description
TA	Trusted Authority
CS_j	Cloud Server
KDC_n	Key Distribution Center
GN_i	General Node
ID_{TA}, RID_{TA}	Identity and pseudo identity of the TA, respectively
ID_{CS_j}, RID_{CS_j}	Identity and pseudo identity of the CS, respectively
ID_{KDC_n}, RID_{KDC_n}	Identity and pseudo identity of the KDC, respectively
ID_{GN_i}, RID_{GN_i}	Identity and pseudo identity of the general node, respectively
G_1,G_2	An additive group and a multiplicative group of prime order q, respectively
g_1	The generator of G_1
D_1, D_2	An additive group and a divisor group of prime order q, respectively
D	Generator of the jacobian group of points of the finite group of points, $J(Fq)$
C	Curve of HECC
$h_0(.)$, $h_1(.)$	SHA-256 and SHA-1 one-way hash functions, respectively
S	Random secret of the TA
Pv_{TA}, Pub_{TA}	Private and public keys of the TA, respectively
Pv_{CS_j}, Pub_{CS_j}	Private and public keys of the CS, respectively
Pv_{KDC_n}, Pub_{KDC_n}	Private and public keys of the KDC, respectively
Pv_{GN_i}, Pub_{GN_i}	Private and public keys of the GN, respectively
$ENC(.)$, $DEC(.)$	Encryption and decryption algorithm for AES-192 encryption, respectively
K, K_1, K_2	Signcryption key and its components, respectively
K_x, K_y	X and Y coordinates of key K, resulting from the mapping function, respectively
$r_i, i \in [1, 2, ..., n]$	Nonce related to the respective situation
$t_i, i \in [1, 2, ..., n]$	Timestamp related to the respective situation
N, Sig	Components of the signature generated by the signcryption
m_i, msg_i, i $\in [1, 2, ..., n]$	Random message related to the respective situation
Δt	Maximum transmission delay
$c_i, i \in [1, 2, ..., n]$	Ciphertext generated for signcryption, in the respective situation
rg_i, $sg, i \in [1, 2, ..., n]$	Nonce, related to the specified situation, and random secret generated for the group key generation phase, respectively
GK_k	Group key of group K.
$M_i, i \in [1, 2, ..., n]$	Message developed related to the respective situation
$E_{k_{x_i}}, i \in [1, 2, ..., n]$	Message encrypted by K_x related to the respective situation
G_k, G_{new}	Respective group of the mentioned situation
G_k, G_{new}	Respective group of the mentioned situation
$data_i$, i $\in [1, 2, ..., n]$	Random data
$Block_i, i \in [1, 2, ..., n]$	Block of the blockchain
B_{sign}	Signature of the block in the blockchain
$BLKHash$, $BlockVer$, $PBLKHash$	Hash, version, and hash of the previous block, of the current block
TS, $Owner$, $Payload$	Timestamp, Owner ID, and payload of the current block
Tx	Transaction of the current block in the blockchain
$T_i, i \in [1, 2, ..., n]$	Average execution time of mentioned element

- Bilinearity: $\forall$ P, Q $\in$ G_1 and $\forall$ a,b $\in$ Z^*_q, e(aP,bQ) = e(aP,bQ) ^{a} = e(aP,bQ) ^{b} = e(aP,bQ) ^{ab}
- Non-degenarate: $\forall$ P, Q $\in$ G_1 such that e(P,Q) $\neq$ 1.
- Computable: $\forall$ P,Q $\in$ G_1 , there is always an effective algorithm to compute e(P,Q).

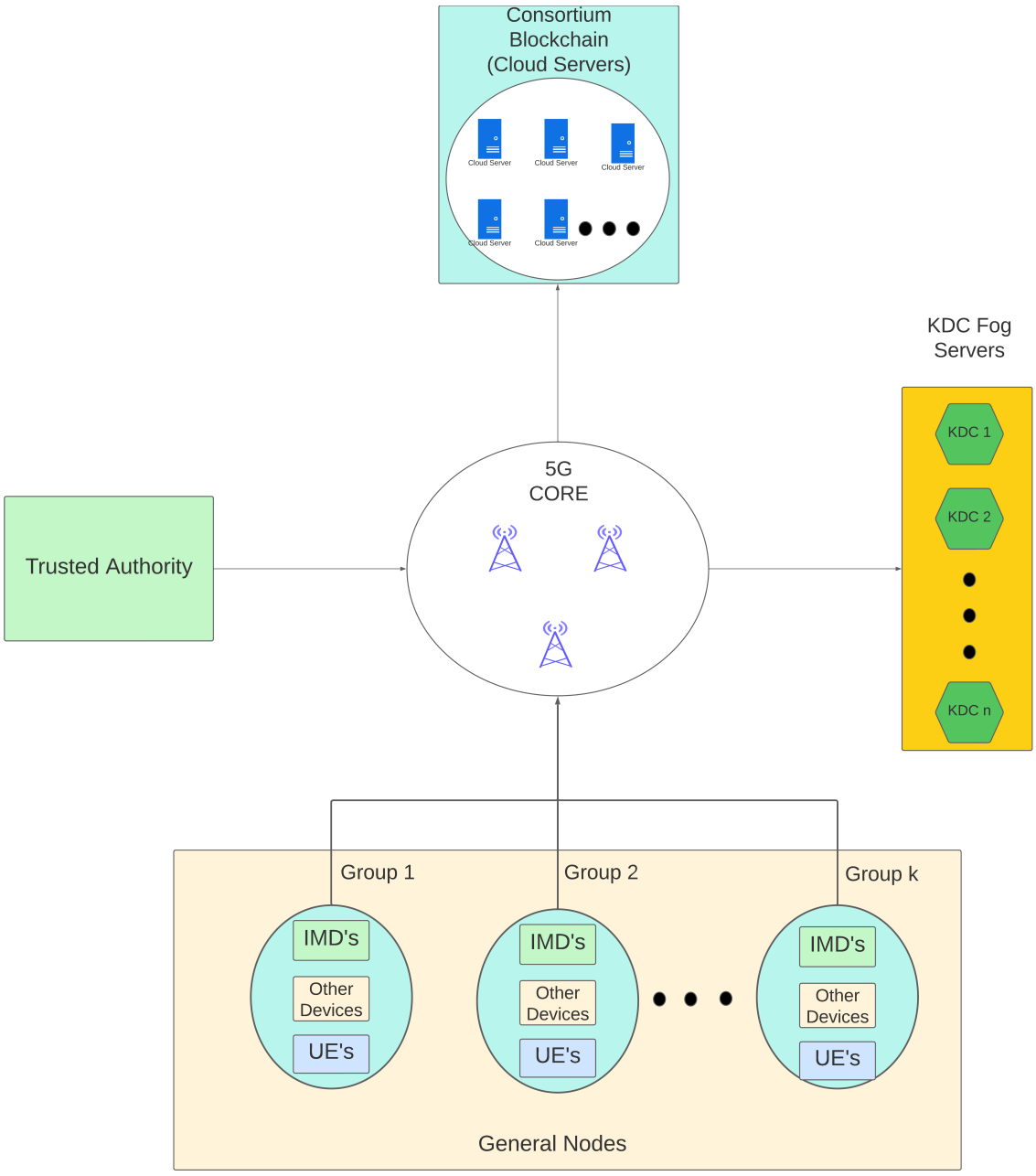

Figure 5: Network architecture of BEAP-IoMT

- Hyper Elliptic Curve Cryptosystem (HECC): Let C be a hyper elliptic curve (HEC) and F_q be a finite field, where q is a large prime number. C is described as y^2 plus $H(x) = f(x)$, where f(x) is a monic polynomial of lower than or equal to $2g + 1$ degree, and H(x) is a polynomial of degree lower than or equal to g degree. A jacobian group of points $J(F_q)$ is created with HEC points and a point at infinity.[19] Then, let D be the generator of $J(F_q)$, which also acts as the official sum of HEC points, represented as:

$$D = (u(x), v(x)) = [\sum_{l=0}^{g} x^l u_l \sum_{l=0}^{g-1} x^l v_l] \tag{III.1}$$

- Hyper Elliptic Curve Digital Signature Algorithm (HEC-DSA): The method introduced in Jian-zhi

et al.[34] was adopted for the creation of a digital signature through HECC encryption, leading to the following sequence of operations:

1. Suppose the existence of a divisor D of HECC operations, a private key x, message abstract h(m),and a mapped function $\theta(Z)$.
2. Generate a random number $k \in F(q)$, in which k is prime with q.
3. Compute $r = k.D$. If $r = [1, 0]$, return to step 2.
4. Map r to $F(q)$, $r \xrightarrow{\theta(Z)} r^{'}$, and calculate $r^{'} = \theta(r) mod(q)$.
5. Compute $s = k^{-1}(h(m) + x.r^{'}) mod(q)$. If s = 0, then return to step 2.
6. Return (r,s) as the digital signature pair.

- The security of the protocol is based on the following computationally infeasible problems:
  - Elliptic Curve Discrete Logarithm Problem (ECDLP): Let E be an elliptic curve over a finite field K. Suppose there are points P, Q ∈ E(K) given such that P is of prime order and Q ∈ <P>. Determine k such that Q = [k]P [40];
  - Hyper Elliptic Curve Discrete Logarithm Problem (HECDL): The hyperelliptic discrete logarithm problem takes on input a hyperelliptic curve of given genus, an element D_1 of the Jacobian, its order n, and another element D_2 in the subgroup generated by $D1$. The problem is to find an integer λ modulo n such that $D_2 = \lambda.D_1$ [26];
  - Computational Diffie-Hellman Problem (CDH): Let a,b ∈ Z^*_q, given g^a and g^b, the CDH problem is to compute g^{ab} [55].
    - (g^a, g^b, g^{ab}), where a and b are random and independent from each other;
    - g^a, g^b, g^c, where a, b, c are random and independent from each other, having c ∈ Z^*_q

III.4 Setup Phase

In this phase, TA defines some common parameters of the system.

- TA chooses the HEC, as in the previous section, and has the following parameters: D,D_1,D_2,q,C.
- TA defines the one-way hash function to be used, $h_0(.)$, $h_0 : 0, 1^* \longrightarrow Z_q^*$ and $h_1(.)$,$h_1 : 0, 1^* \longrightarrow Z_q^*$, as a SHA-256 and SHA-1, respectively.
- TA generates its own ID, defined as ID_{TA}, and a pseudo identity RID_{TA} using secret key S of their own making, as RID_{TA}=$h_1(ID_{TA}||S)$,
- A random private key $Pv_{TA} \in Z_q^*$ is generated for the trusted authority. It then uses the newly created key to compute its public key as $Pub_{TA} = Pv_{TA}$. D.
- Encryption and Decryption algorithms, ENC(.) and DEC(.), respectively, will be set by the TA.
- TA has the following parameters: [D_1,D_2,D,q,C,R,$h_0(.)$, $h_1(.)$,ENC(.),DEC(.),ID_{TA},RID_{TA}].

III.5 Registration Phase

1. Cloud Server (Miner) Registration: TA chooses a unique identity ID_{CS_j} for all j cloud servers and computes a pseudo-identity for them, $RID_{CS_j} = h_1(ID_{CS_j}||S)$, where S is the TA's secret key. Private key Pv_{CS_j} of the miners is generated by the cloud server, and public key $Pub_{CS_j} = Pv_{CS_j}.D$ will computed by the trusted authority. TA then stores [RID_{CS_j},RID_{TA},D,D_1,D_2, Pub_{CS_j},Pv_{CS_j},ENC(.),DEC(.),$h_0(.), h_1(.)$] in the cloud server's database.

2. Key Distribution Centers: KDCs have a unique identity set by TA, ID_{KDC_n} for all n fog servers. TA then computes pseudo identity $RID_{KDC_n} = h_1(ID_{KDC_n}||S)$ and KDC produces private key Pv_{KDC_n}. Public key $Pub_{KDC_n} = Pv_{KDC_n}.D$ is generated. TA then applies [RID_{KDC_n},Pub_{KDC_n}, Pv_{KDC_n},$h_0(.)$,$h_1(.)$,ENC(.),DEC(.),D,D_1,D_2] to the KDC storage and sends Pub_{KDC_n} to the cloud servers. After authentication with the cloud servers, the KDC retrieves the most up-to-date list of circular nodes in the network.

3. General Node Devices: The new devices have a unique identity set by TA, ID_{GN_i}. The authority then computes pseudo identity $RID_{GN_i} = h_1(ID_{GN_i}||S)$. Private key Pv_{GN_i} and public key $Pub_{GN_i} = Pv_{GN_i}.D$ are computed. Finally, TA stores [RID_{GN_i},Pub_{GN_i}, Pv_{GN_i},$h_0(.)$,$h_1(.)$,ENC(.),DEC(.),D,D_1,D_2] in the GN local storage and sends Pub_{GN_i} to the cloud server.

4. During registration, the GNs can choose to join a group or create a new one. All GNs upon registration, are arranged in a circular list, L, according to their identities and group id.

5. Public keys are available for all devices in the network.

III.6 Mutual Authentication between Cloud Servers and KDCs

In this phase, every KDC must be properly authenticated with the cloud servers through key sharing and session establishment. An overview of the phase can be seen in figure 6.

1. Having access to the public keys of KDCs via thr aforementioned secure channel, CS_j computes a nonce r_{CS_j}.

2. CS_j generates key K = $K_1 + K_2$, such that $K_1 = r_{CS_j} \times ID_{CS_j} \times D$ and $K_2 = r_{CS_j} \times ID_{CS_j} \times (D + Pub_{KDC_n})$.

3. The HECC key consists of points in x and y coordinates, usually represented by P, and expressed as $K = P(x, y)$; then, employing a polynomial integer mapping function (IPM), this key can be expressed as $K = (K_x, K_y)$ [19].

4. The miner produces a timestamp t_1.

5. CS_j calculates ciphertext $c_1 = ENC(m_1, t_1)$, where m_1 is a random message.

6. The cloud server computes $N_1 = h_0(m||K_x||t_1)$ and $Sig = [(r_{CS_j} \times ID_{CS_j}) - (Pv_{CS_j} \times N_1)] \times D$ to serve as the signcryption signature.

7. CS_j sends (c_1,N_1,Sig,t_1) to the corresponding KDC.

   KDC must now perform the unsigncryption of the tuple from CS_j.

8. Firstly, KDC_n generates a current timestamp $t_1^{'}$ and verifies $|t_1^{'} - t_1| \leq \Delta t$. If it is true, authentication proceeds; otherwise, the connection with CS_j is terminated, and authentication fails.

9. KDC_n computes key $K = K_1 + K_2$, where $K_1 = Sig + (N \times Pub_{CS_j}$ and $K_2 = K_1 \times (1 + Pv_{KDC_n})$.

10. The same IPM used by the miner must be applied again for K, such that $K = (K_x, K_{y)}$.

11. $DEC(c_1) = (m_1, t_1)$ is calculated.

12. KDC_n computes $N_1^{'} = h_0(m_1||K_x||t_1)$ and verifies $N_1^{'} = N_1$. If it is true, the cloud server is authenticated by KDC; otherwise, authentication fails and connection is terminated.

13. The fog server generates a random message m_2, current timestamp t_2, ciphertext $c_2 = ENC(m_2, t_2)$ and $N_2 = h_0(m_2||K_x||t_2)$.

14. KDC_n sends tuple (c_2,N_2,t_2) to CS_j.
    Finally, the cloud server must authenticate the KDC.

15. CS_j generates current timestamp $t_2^{'}$ and verifies $|t_2^{'} - t_2| \leq \Delta t$. If it is true, authentication proceeds; otherwise, the session is terminated and authentication fails.

16. $DEC(c_2) = (m_2, t_2)$ is computed.

17. Finally CS_j computes $N_2^{'} =_0 h(m_2||K_x||t_2)$ and verifies $N_2^{'} = N_2$. If it is true, authentication proceeds; otherwise, authentication fails and connection is terminated.

III.7 General Node Authentication

Suppose $j \neq i \neq l$, and $i, j, k \in L, where L \in Z_q^*$.

1. GN_i computes timestamp $t_{1_{GN_i}}$ and the node then broadcasts (Pub_{GN_i},$t_{1_{GN_i}}$) to all other nodes and the corresponding KDC.

2. GN_i receives the message (Pub_{GN_j},$t_{1_{GN_j}}$) for all other j nodes.

3. GN_i creates a current timestamp $t_{1_{GN_{ij}}}^{'}$, and verifies $|t_{1_{GN_{ij}}}^{'} - t1_GN_j| \leq \Delta t$. If it is true, authentication proceeds; otherwise, the authentication error message is broadcast, warning the network of the failed validation of node GN_j.

4. GN_i chooses a nonce $r_{ij} \in Z_q^*$ and produces a random message m_{ij}.

5. GN_i generates key K_{ij} = $K_{1_{ij}} + K_{2_{ij}}$, such that $K_{1_{ij}} = r_{ij} \times ID_{GN_i} \times D$ and $K_{2_{ij}} = r_{ij} \times ID_{GN_i} \times (D + Pub_{GN_j})$.

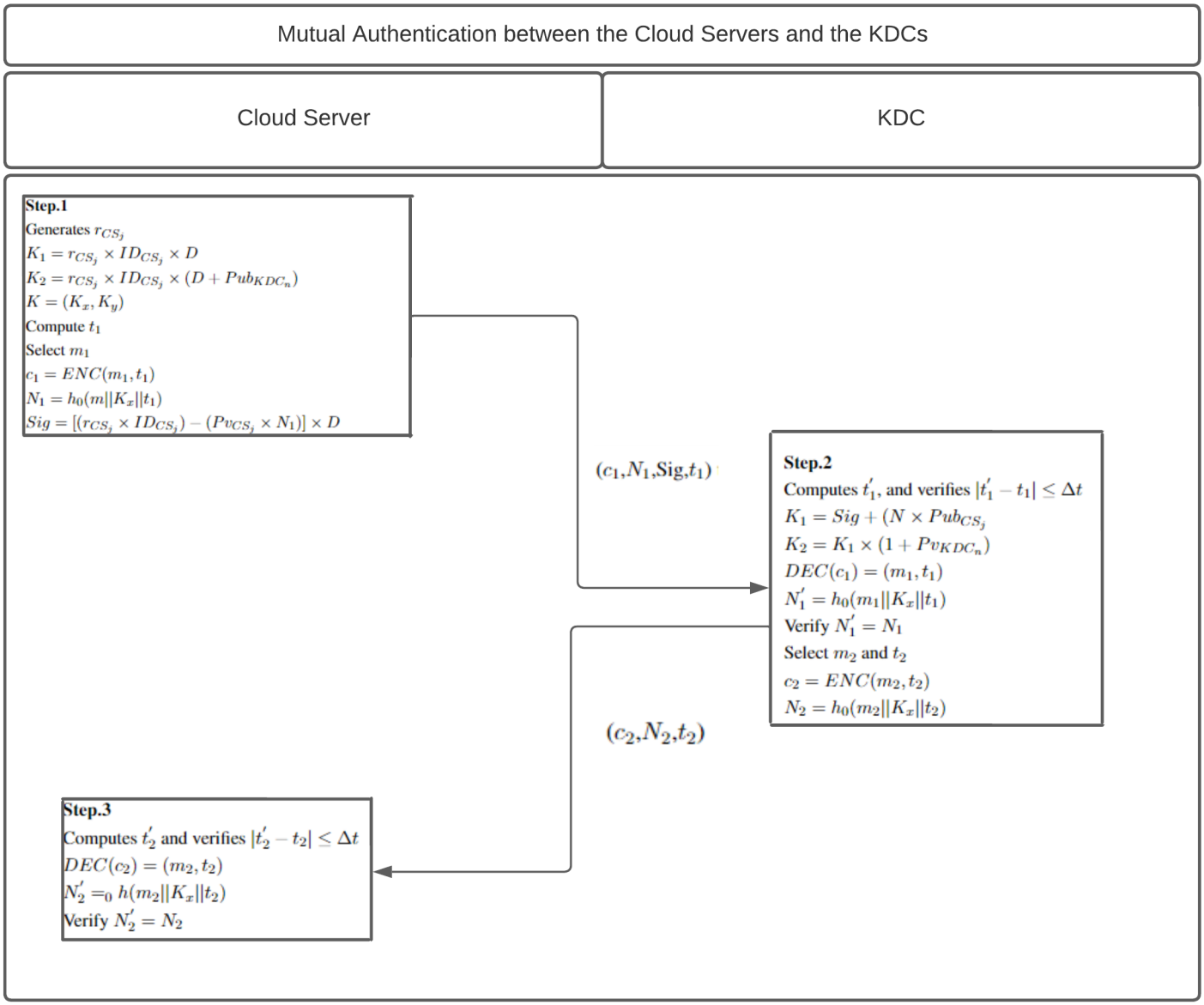

Figure 6: Summary of Mutual Authentication between the Cloud Servers and the KDCs phase.

6. The use of a polynomial integer mapping function (IPM) (see sub-section III.VI) enables the obtaining of key $K_{ij} = (K_{ij_x}, K_{ij_y})$.

7. The current node produces timestamp $t_{2_{ij}}$.

8. GN_i calculates ciphertext $c_{ij} = ENC(m_{ij}, t_{2_{ij}})$.

9. GN_i computes $N_{ij} = h_0(m_{ij}||K_{x_{ij}}||t_{2_{ij}})$ and $Sig_{ij} = [(r_{ij} \times ID_{GN_i}) - (Pv_{GN_i} \times N_{ij})] \times D$, to act as the signcryption signature.

10. GN_i sends (c_{ij},N_{ij},Sig_{ij},$t_{2_{ij}}$) to GN_j.

11. GN_i receives (c_{ki},N_{ki},Sig_{ki},t_{ki}).

12. GN_i computes current timestamp t_{ik}, and verifies $|t_{ik} - t_{ki}| \leq \Delta t$. If it is not true, it broadcasts an

authentication failed message regarding GN_k. At this moment the current node must authenticate the message received from GN_k, by unsigncryption.

13. GN_i computes key $K_{ik} = K_{1_{ik}} + K2_{ik}$, where $K_{1_{ik}} = Sig + (N \times Pub_{Gn_k}$ and $K_{2_{ik}} = K_{1_{ik}} \times (1 + Pv_{Gn_i})$.
14. The same IPM used by the signer must be applied again for K, such that $K_{ik} = (K_{x_{ik}}, K_{y_{ik}}$.
15. $DEC(c_{ki}) = (m_{ki}, t_{ki})$ is calculated.
16. GN_i computes $N_{ik} = h_0(m_{ki}||K_{x_{ki}}||t_{ki})$ and verifies $N_{ik} = N_{ki}$. If it is true, GN_k is authenticated by GN_i; otherwise, the connection is terminated and an authentication failed message is broadcast.

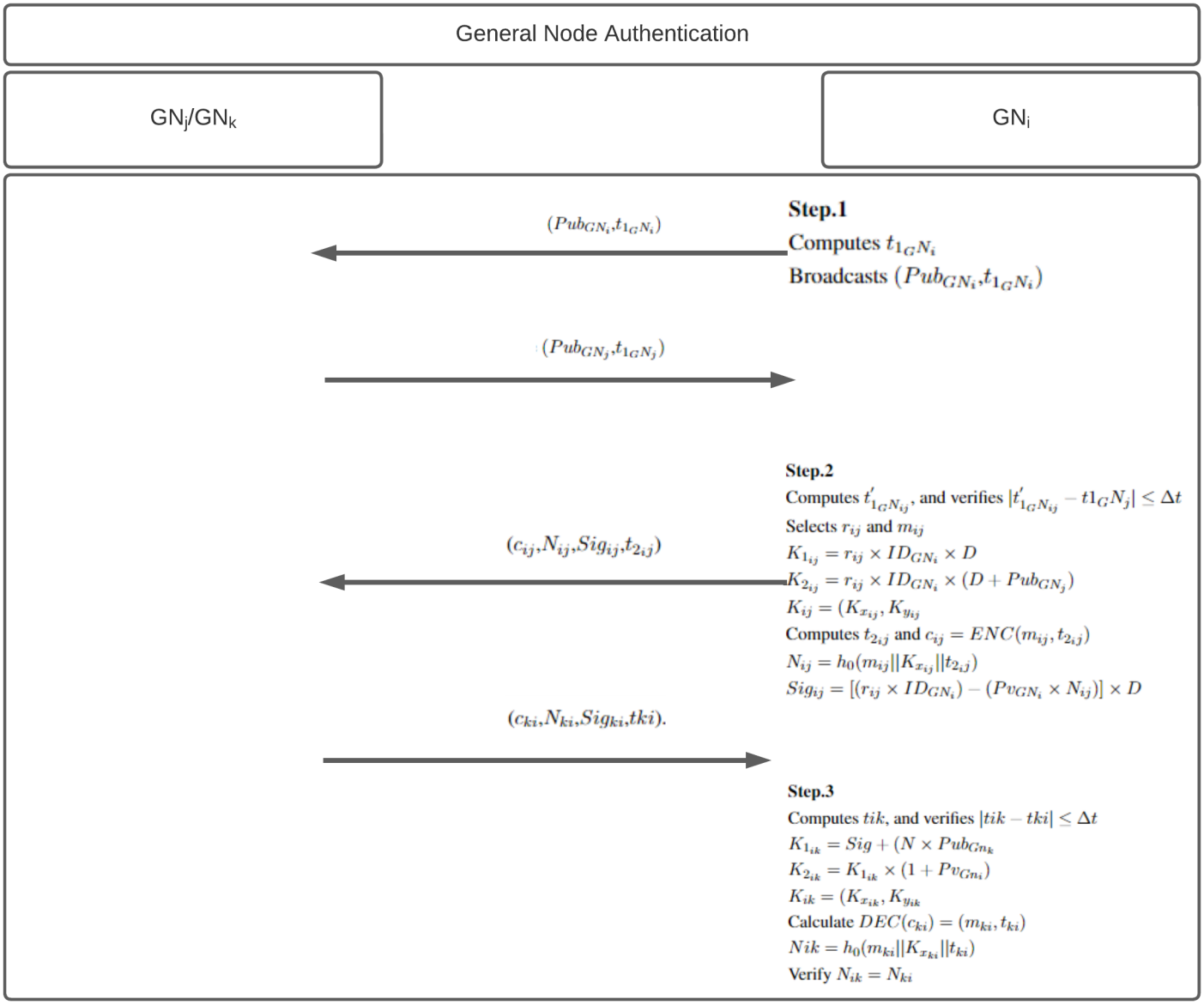

Figure 7: Summary of General Node Authentication phase.

After the General Nodes have authenticated each other, a batch authentication must be performed via KDC. An overview of the phase can be seen in figure 7.

III.8 Batch Authentication

This phase describes the batch of authentication of the many group members by the KDC. An overview of the phase can be seen in figure 8.

1. All general nodes can access the public key of KDC via either TA during registration, or a request towards an already trusted KDC.
2. GN_i chooses a nonce $r_{in} \in Z_q^*$ and produces a random message m_{in}.
3. GN_i generates key K_{in} = $K_{1_{in}} + K_{2_{in}}$, such that $K_{1_{in}} = r_{in} \times ID_{GN_i} \times D$ and $K_{2_{in}} = r_{in} \times ID_{GN_i} \times (D + Pub_{KDC_n})$.
4. The use of a polynomial integer mapping function (IPM) (see sub-section III.VI) enables the obtaining of key $K_{in} = (K_{x_{in}}, K_{y_{in}})$.
5. GN_i produces timestamp t_i.
6. GN_i then calculates ciphertext $c_{in} = ENC(m_{in}, t_i)$.
7. GN_i computes $N_{in} = h_0(m_{in}||K_{x_{in}}||t_i)$ and $Sig_{in} = [(r_{in} \times ID_{GN_i}) - (Pv_{GN_i} \times N_{in})] \times D$, to act as the signcryption signature.
8. GN_i sends (c_{in},N_{in},Sig_{in},tin) to the respective KDC.
   A batch unsigncryption of the interested nodes is then performed.
9. KDC_n receives (c_{in},N_{in},Sig_{in},tin) and computes a timestamp $t_{in}^{'}$ to validate $|t_{in}^{'} - t_{in}| \leq \Delta t$. If it is true, KDC proceeds with authentication; otherwise, an authentication error message is broadcast to all i nodes upon receiving the message.
10. Having all messages, KDC_n computes $Nt_{in} = \sum_{i=0}^{l} N_{in}$.
11. For all messages, the fog server calculates key $K_{in} = K_{1_{in}} + K2_{in}$, where $K_{1_{in}} = Sig_{in} + (N \times Pub_{GN_i}$ and $K_{2_{in}} = K_{1_{in}} \times (1 + Pv_{KDC_n})$.
12. The same IPM used by the signer must be applied for K, such that $K_{in} = (K_{x2_{in}}, K_{y2_{in}}$.
13. $DEC(c_{in}) = (m_{in}, t_{in})$ is calculated.
14. KDC_n computes $N_{in}^{'} = h_0(m_{in}||K_{x2_{in}}||t_{in})$.
15. The fog server then computes $Nt_{in}^{'} = \sum_{i=0}^{l} N_{in}^{'}$, and verifies, $Nt_{in}^{'} = Nt_{in}$. If it is true, all l nodes are authenticated by KDC_n; otherwise, it broadcasts an authentication failure message.
16. The fog server generates a random message m_{ni}, current timestamp t_{ni}, ciphertext $c_{ni} = ENC(m_{ni}, t_{ni})$ and $N_{ni} = h_0(m_{ni}||K_{x2_{in}}||t_{ni})$.
17. KDC_n broadcasts tuple (c_{ni},N_{ni},t_{ni}) to the general nodes.
    The general nodes must authenticate KDC to finish authentication.

18. GN_i generates current timestamp $t_{ni}^{'}$ and verifies $|t_{ni}^{'} - t_{ni}| \leq \Delta t$. If it is true, it proceeds with the authentication. Otherwise the session is terminated and authentication fails.

19. $DEC(c_{ni}) = (m_{ni}, t_{ni})$ is calculated.

20. GN_i computes $N_{ni}^{'} = h_0 h(m_{ni}||K_{x_{in}}||t_{ni}^{'})$, and verifies $N_{ni}^{'} = N_{ni}$. If it is true, the fog server is authenticated by the miner; otherwise, authentication fails and connection is terminated. If it is true, KDC is authenticated by the generic node and the scheme can proceed to the group key generation phase. Otherwise, it broadcasts an authentication failure message.

21. GN_i computes $m_{in}^{'} = ENC(ID_{GN_i})$ using $K_{x_{in}}$ to encrypt the ID.

22. GN_i selects a current timestamp t_{id} and sends $(m_{in}^{'}, t_{id})$ to the fog server.

23. The Fog server computes a current timestamp $t_{id}^{'}$ and validates $|t_{id}^{'} - tid| \leq \Delta t$.If it is false, the connection is terminated.

24. KDC_n extracts ID_{GN_i} using $K_{x2_{in}}$.

III.9 Group Key Generation Phase

This phase describes the process of the creation of the group key. An overview of the phase can be seen in figure 9.

1. KDC_n generates a nonce rg_n and a random secret $sg \in Z_q^*$.

2. KDC_n derives group key GK_k for nodes GN_1,GN_2,...,GN_l, members of group G_k, where l corresponds to the values present in list L, using its secret key and information received from members of the group, such as $GK_k = h_0(ID_{GN_1}||ID_{GN_2}||...||ID_{GN_l}||rg_n||sg||PV_{KDC_n})$.

3. KDC_n computes current timestamp tg_n.

4. KDC encrypts the group key for all legitimate members of G_k, using $K_{x_{in}}$ created for each group node by computing $E_{k_{x_{ni}}} = ENC(GK_k||tg_n)$.

5. KDC_n sends message $M_i = (E_{k_{x_{ni}}}, N_{in}^{'}, tg_{n)}$ to all valid members.

6. Upon receiving M_i, GN_i generates timestamp $tg_n^{'}$ and validates $|tg_n^{'} - tgn| \leq \Delta t$. If it is true, it checks the authenticity of the sender.

7. GN_i decrypts $E_{k_{x_{ni}}} = ENC(GK_k||tg_n)$.

8. GN_i validates $N2_{in} = h_0(GK_k||K_{x_{in}}||tg_n) = N_{in}^{'}$. If it is true, the message is authenticated and the group member stores key GK_k, since it is valid. Otherwise, it is discarded and a failure message is broadcast.

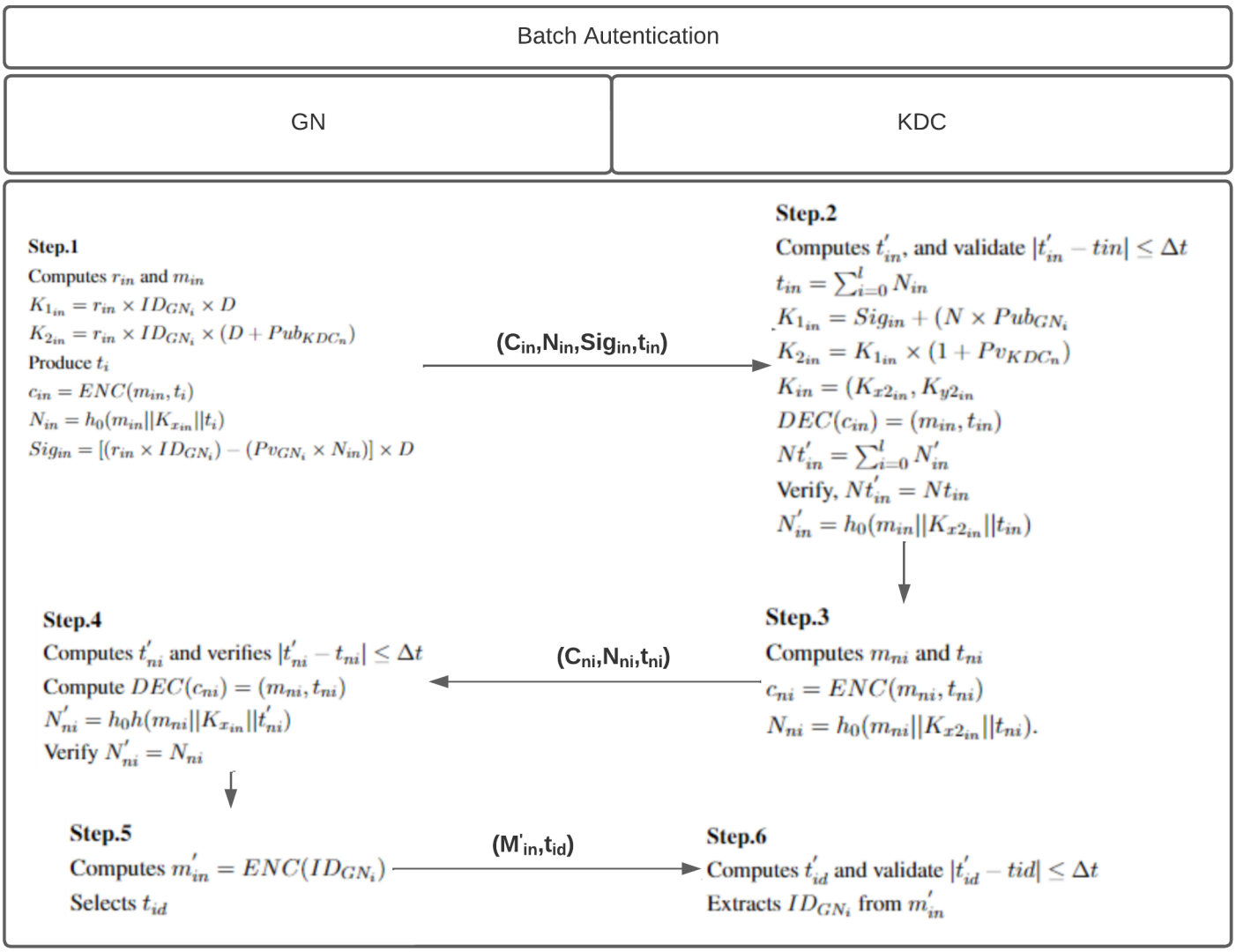

Figure 8: Summary of Batch Authentication phase.

9. Towards ending the group key generation phase, KDC_n sends group key GK_k to the blockchain after encrypting it with the key computed and agreed upon (see section 4.3.6), together with a newly computed timestamp t_{GK_k} as $c_{GK_k} = ENC(GK_k, t_{GK_k})$, and sends (c_{GK_k}, t_{GK_k}) to the miner cloud server.

10. The cloud server computes a timestamp t'_{GK_k} upon receiving (c_{GK_k}, t_{GK_k}) from the fog server, calculating $|tg'_{GK_k} - tg_{GK_k}| \leq \Delta t$. If is true, it decrypts and extracts GK_k, and stores it in the blockchain for the respective group.

III.10 Group Join Phase

When a new GN, referred to as GN_{new} wants to join an existing group, be it a node that wishes to change groups, or a newly registered node, it must be validated by the corresponding KDC.

1. GN_{new}retrieves list L from the blockchain, adds its identity to it, and broadcast it to the desired

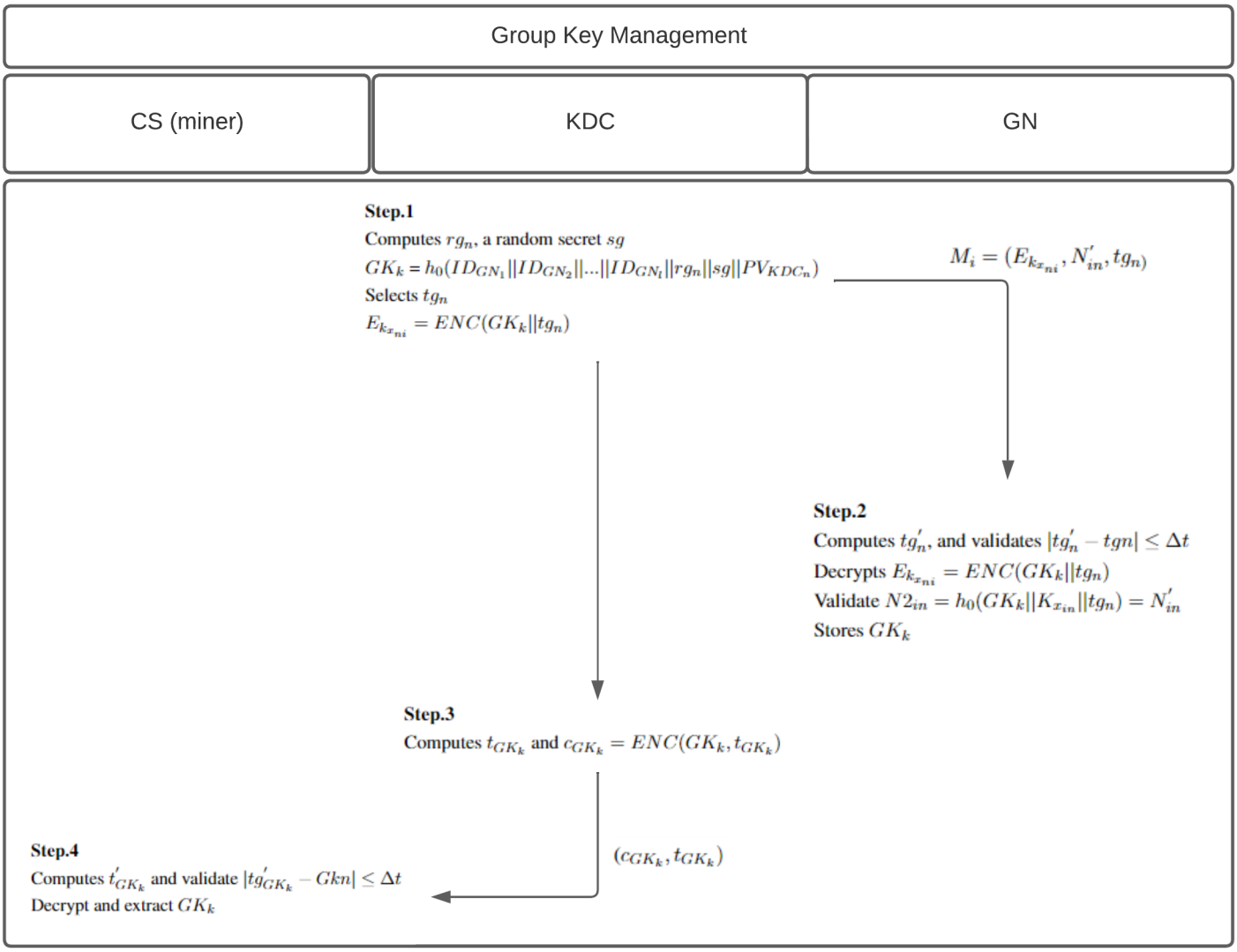

Figure 9: Summary of Group Key Management phase.

group.

2. GN_{new} retrieves the list L from the blockchain and adds its identity to the list L, L, thus being broadcast to the desired group.

3. GN_{new} computes timestamp t_{1new} and the node broadcasts ($Pub_{GN_{new}}$,t_{1new}) to all other nodes and the corresponding KDC.

4. After receiving the public key, all other j nodes belonging to list L compute timestamp t'_{1new} and validate $|t'_{1new} - t_{1new}| \leq \Delta t$. If it is true, they send (Pub_{GN_j},t_{1_j}) to the node requesting membership. Otherwise, they broadcast an authentication failure message to the rest of the group.

5. GN_{new} creates a current timestamp t'_{1_j}, and verifies $|t'_{1_j} - t_{1_j}| \leq \Delta t$. If it is true, it proceeds with authentication. Otherwise, an authentication error message is broadcast, warning the network of the failed validation of node GN_j.

6. GN_{new} chooses a nonce $r_{new} \in Z_q^*$ and produces a random message m_{new}.

7. GN_{new} generates key K_{new} = $K_{1_{new}} + K_{2_{new}}$, such that $K_{1_{new}} = r_{new} \times ID_{GN_{new}} \times D$ and $K_{2_{new}} = r_{new} \times ID_{GN_{new}} \times (D + Pub_{GN_j})$.

8. The use of a polynomial integer mapping function (IPM) (see sub-section III.VI), enables the obtaining of key $K_{new} = (K_{x_{new}}, K_{y_{new}})$.

9. The current node produces timestamp t_{2_new}.

10. GN_{new} then calculates ciphertext $c_{new} = ENC(m_{new}, t_{2_new})$.

11. GN_{new} computes $N_{new} = h_0(m_{new}||K_{x_{new}}||t_{2_new})$ and
$Sig_{new} = [(r_{new} \times ID_{GN_{new}}) - (Pv_{GN_{new}} \times N_{new})] \times D$ to act as the signscryption signature.

12. GN_{new} sends (c_{new},N_{new},Sig_{new},t_{2_new}) to GN_j.

13. GN_j receives (c_{new},N_{new},Sig_{new},t_{2_new}).

14. GN_j computes current timestamp $t^{'}_{2new}$ upon receiving the message and verifies $|t^{'}_{2new} - t_{2new}| \leq \Delta t$. if it is not true, it broadcasts an authentication failure message on GN_{new} to the group.

15. GN_j computes key $K_j = K_{1_j} + K2_j$, where $K_{1_j} = Sig + (N \times Pub_{Gn_{new}}$ and $K_{2_j} = K_{1_j} \times (1 + Pv_{Gn_j})$.

16. The same IPM used by the signer must be applied for K, such that $K_j = (K_{x_j}, K_{y_j}$.

17. $DEC(c_{new}) = (m_{new}, t_{2_new})$ is calculated.

18. GN_j computes $Nj = h_0(m_{new}||K_{x_j}||t_{2_new})$ and verifies $N_j = N_{new}$. If it is true, GN_{new} is authenticated by GN_j. Otherwise, the connection is terminated and an authentication failed message is broadcast.

19. GN_j computes a timestamp $t2_j$ and a message m_j and encrypts them with K_{x_j}, as $E_{k_{x_j}} = ENC(m_j||t2_j)$.

20. GN_j generates $N^{'}_j = h_0(m_j||K_{x_j}||t2_j)$

21. GN_j sends message $M_j = (E_{k_{x_j}}, N^{'}_j, t2_{j)}$ to all valid members.

22. Upon receiving M_j, GN_{new} generates timestamp $t2^{'}_j$, and validates $|t2^{'}_j - t2_j| \leq \Delta t$. If it is valid, it checks the authenticity of the sender.

23. GN_{new} decrypts $E_{k_{x_j}} = ENC(m_j||t2_j)$,with its K_x.

24. GN_i validates $N2_{new} = h_0(m_j||K_{x_{new}}||t2_j) = N^{'}_j$. If it is true, the message is authentic and G_j is valid.

25. KDC repeats the Batch Authentication phase, now including G_{new}.

26. The network repeats the steps of the Group Key Generation Phase with the added element of G_{new} and excludes the current group key, thus concluding the group key update.

III.11 Group Exit Phase

Whenever a GN_j desires to leave its group, it updates list L and distributes the new information.

1. GN_j deletes its identity from the list and broadcasts the new L to all members of the the new list, including KDC.
2. KDC notifies the blockchain on the removal of the node.
3. General Node Authentication and Batch Authentication phases are repeated for the group members.
4. A new group key is generated via Group Key Generation Phase and the current one is discarded.

IV BLOCKCHAIN CONSTRUCTION PHASE

This section describes the steps for the construction of the blockchain and its initial configurations.

1. Whenever node GN_i must send data to fog server KDC_n, it generates a timestamp t1, encrypts a message msg_1 using group key GK_k, as $msg_1 = ENC(data_i, t1)$, and sends it to the server via an open channel.
2. Upon receiving msg_1,the Fog server decrypts it with the group key it has stored, generates a timestamp $t^{'}1$, and verifies $|t^{'}1 - t1| \leq \Delta t$. If it is true, KDC_ngenerates a new timestamp t2 and encrypts the message with the key established for communication with cloud server CS_j (see Section 4.3.6), as ciphertext $c_1 = ENC(data_1, t2)$, using key K_x from the signscryption, via its signature.
3. KDC_n sends c_1 to CS_j via open channel.
4. Upon receiving c_1, the miners (cloud servers) decrypt it with their secret key for communication with KDC_n, and extract $(data_1, t2)$ and calculate $|t^{'}2 - t2| \leq \Delta t$, where $t^{'}2$ is the timestamp generated by CS_j after receiving the ciphertext. If it is true, c_1 and its data, $data_1$, are classified as a valid message.

Using $data_1$, CS_j starts the block creation process, followed by its addition to the blockchain. A block is added to the chain if it has been successfully validated by the other cloud servers via the XRP ledger consensus algorithm, in the Peer-to-Peer (P2P) network, where the miners belong.

The concept of a consortium and private blockchains provides secrecy, confidentiality, added security, and a higher degree of control of the blockchain. The following steps are taken for the creation of a block belonging to the chain, i.e., $Block_i$:

1. After collecting $data_1$, CS_j begins their transactions. Suppose $T_{x1}, T_{x2}, ..., T_{x_{nt}}$ represent those transactions, where $n \in Z_q^*$.
2. A regular public key, PUB_{CS_j},is used here for the encryption mechanism through HECC, once its secret values had been stored in the cloud servers since the registration phase.

3. $ENC_{Pub_{CS_j}}(Tx_i)$, where $i = [1, 2, ..., nt]$, stands for the encrypted transactions that will be used for computing the "Merkle Tree", by building the Merkle Tree Root (MTR), since it is a fundamental element of the construction of the hash chain in the blockchain.

4. Hash of information [BlockVer,PBLKHash,MTR,TS,Owner,Payload], is computed for the current hash block, BLKHash.

5. Following the properties described in Sections 1.3.2 and 1.3.3, a Hyper Elliptic Curve Digital Signature Algorithm (HEC-DSA) produces signature $Bsign = (r_{Block_i}, s_{Block_i})$ in message $msg = BLKHash$ for the current block, which will be validated by HEC-DSA verification algorithm ([51]).

6. With $Block_i$ built by CS_j, it is forwarded to the other miner nodes in the network.

The following steps are required for both verification and addition of the block by the consensus algorithm:

1. Whenever a block $Block_i$ is received by a cloud server, an external miner is chosen as a leader L, via the leader selection process in algorithm 1 (Figure 10), based on Zhang et al.[56]. The following algorithms are based on the ones proposed by Garg [25]:

2. The voting mechanism in XRP Ledger [17] is used for consensus of the chain; all cloud servers have a HECC private-public key pair (Pv_{CS_j}, Pub_{CS_j}) and the public keys of the cloud servers are known by each other.

3. The initial values for input for the initialization of the consensus algorithm are:
   Input: $Block_i$ is initialized in the miner as $Block_i = [BlockVer, PBLKHash, MTR,$
   $t_j, Owner, Payload, Pub_{CS_j}, [ENC_{Pub_{CS_j}(Tx_s}|s = 1, 2, ..., n_t], BLKHash, BSign.$
   Private-public key pairs $[Pv_{CS_j}.Pub_{CS_j} = Pv_{CS_j}.D]$ are set for all miners in the network.

   Algorithm 2 refers to the detailing of the output of the consensus process, as seen in Garg [25], regarding the RPCA consensus.

4. Suppose a user, identified by its RID and its group identification (GID) – if it has one - wishes to access secret information such as $data_u$, from $Block_i$. It, therefore, must access its corresponding fog server, or directly the cloud server, where its request will be processed.

5. Either KDC goes to the corresponding cloud server, or the user communicates directly with the miner node, $CS_j^{'}$. Since $CS_j^{'}$ shares a common ledger with CS_j, and with other members of the blockchain; it has access to $Block_i$, hence, $data_u$, via extraction with blockchain operations.

6. Supposing $CS_j^{'}$ extracts requested data $data_u$, the cloud server will also apply hash function $h_0(.)$ to the collected value, as $h_0(data_u)$. $CS_j^{'}$ extracts $ENC_{GK_k}(h_0(data_u^{'}))$ as new data to be used for its knowledge of divisor D to obtain $h_0(data_u^{'})$ and compares $h_0(data_u) = h_0(data_u^{'})$. If it is true, the block is not modified and is considered trusted. Otherwise, the cloud server discards the data of $Block_i$.

7. Having both group key and a respective key to KDC, $CS_j^{'}$ communicates with the user, regardless of its origin.

Input: $Block_i$ is initialized in the miner,as $Block_i = [BlockVer, PBLKHash, MTR, t_j, Owner, Payload, Pub_{CS_j}, [ENC_{Pub_{CS_j}(Tx_s}|s = 1, 2, ..., n_t], BLKHash, BSign.$
Private-public key pairs [$Pv_{CS_j}.Pub_{CS_j} = Pv_{CS_j}.D$];
$CS_i \rightarrow Follower(i \in 1, 2, ..., n)$, where n $> 3f + 1$, f is the fault nodes number;
Set the tenure number to 0, $TN_{CSi} = 0$ $(i \in 1, 2, ..., n)$;
Set the original number of votes to 0, $N_v = 0$;
Start the Timer, set a random timeout T_{out} ;
while $Timer > T_{out}$ **do**
    $Follower \rightarrow Candidate$;
    $TN + 1$;
    Start the timer;
    $N_v + 1$
    Send a request of voting to all other nodes and wait for the reply votes;
    if Receive votes reply **then**
        Compute N_v again;
        if $N_v > \frac{n}{2} + 1$, where n is the total nodes number **then**
            $Candidate \rightarrow Leader$;
        end if
    else if Receive Leader confirmed **then**
        $Candidate \rightarrow Follower$;
    else
        Repeat steps 7-10 for a new election;
    end if
end while

Figure 10: Algorithm 1 - Leader Selection Procedure

8. $CS_j^{'}$ generates a current timestamp t2 and computes $msg = ENC_{CS_j^{'}}(data_1, t2)$, by encrypting it with the extracted group key, or the shared key for connection with the fog server; the choice of the key is depends if the current group key has already been estabilished, if it has already been created, the group key is used for communication, otherwise the blockchain is still able to communicate with the fog servers by the shared key between them; this process is important, since communication between the fog server and the blockchain can occur with no group key already established.

9. The cloud server sends msg to the user who decrypts it and extracts $data_1$ and $t2$. The user validates the timestamp via a newly generated $t^{'}2$, as $|t^{'}2 - t2| \leq \Delta t$.If it is true, the message is valid; otherwise, it is discarded and connection is terminated.

General nodes and fog servers can access the desired data with the built blockchain and keep themselves updated with the transaction information.

V DYNAMIC NETWORK DEVICE ADDITION

V.1 General Node Dynamic Addition

The protocol enables a future addition of new general nodes to the network.

1. TA chooses a unique identity $ID_{GN_{new}}$, for new node GN_{new} and computes $RID_{GN_{new}} = h_1(ID_{GN_{new}}||S)$, public key $Pub_{GN_{new}} = RID_{GN_{new}}.D$, and private key $Pv_{GN_{new}}$
2. TA stores [$RID_{GN_{new}}$,$Pub_{GN_{new}}$,$Pv_{GN_{new}}$,$h_0(.)$,$h_1(.)$,ENC(.),DEC(.),D,D_1,D_2] in the GN local storage and sends $Pub_{GN_{new}}$ to the cloud servers.

V.2 Fog Server Dynamic Addition

As addressed in the previous section, the fog servers can communicate with the blockchain; therefore, whenever a fog server is removed, out of order, or simply updated by a newly added KDC, the server contacts the chain and extracts data necessary for its acting on the network.

1. TA chooses a unique identity $ID_{KDC_{new}}$ for a new fog server KDC_{new}, computes $RID_{KDC_{new}} = h_1(ID_{KDC_{new}}||S)$ and public key $Pub_{KDC_{new}} = RID_{KDC_{new}}.D$.KDC produces private key $Pv_{KDC_{new}}$.
2. TA applies [$RID_{KDC_{new}}$,$Pub_{KDC_{new}}$,$Pv_{KDC_{new}}$,$h_0(.)$,$h_1(.)$,ENC(.),DEC(.),D,D_1,D_2] on the KDC storage, and sends $Pub_{KDC_{new}}$ to the cloud servers. After authentication with the cloud servers, KDC retrieves the most up-to-date list of circular nodes in the network.
3. If a fog server becomes unavailable, the authority of current and future groups of its vicinity is sent to another server via a request from the new KDC to the blockchain on the list containing group members and other important data and proceeds to a new authentication phase. It can also retrieve any important information on specific members of the group.

V.3 Cloud Server Dynamic Addition

The scheme also enables a future addition of new cloud servers to the infrastructure and miners to the blockchain.

1. TA chooses a unique identity $ID_{CS_{new}}$ for a new fog server CS_{new}, computes $RID_{CS_{new}} = h_1(ID_{CS_{new}}||S)$ and public key $Pub_{CS_{new}} = RID_{CS_{new}}.D$. The miner produces private key $Pv_{CS_{new}}$.
2. The trusted authority stores [$RID_{CS_{new}}$,RID_{TA},D,D_1,D_2,$Pub_{CS_{new}}$,$Pv_{CS_{new}}$,ENC(.),DEC(.), $h_0(.), h_1(.)$] in the cloud server's database.

VI SECURITY ANALYSIS AND FUNCTIONALITIES COMPARISON

VI.1 Correctness Proof

This section discusses the correctness of the signcryption method used for the protocol (see[[19]]).

Considering that all entities in the network model of BEAP-IoMT, at some point must realize the signcryption process at some point in the authentication process and the necessary steps for operating the signcryption and unsigncryption phases of the process; suppose that the singcryption operates via the operations of a sender and an addressee, the first, who executes the signcryption phase and sends to the addressee, who executes the unsigncryption phase. We can analyse the correctness of the signcryption process:

$K_1 = Sig + (N \times Pub_{sender}) = [(r_{sender} \times ID_{sender}) - (Pv_{sender} \times N)] \times D + (N \times Pub_{sender}) = D \times r_{sender} \times ID_{sender} - D \times (Pv_{sender} \times N) + N \times Pub_{sender} = D \times r_{sender} \times ID_{sender} - N \times Pub_{sender} + N \times Pub_{sender} = D \times r_{sender} \times ID_{sender} = K_1$

$K_2 = K_1 \times (1 + Pv_{addressee}) = D \times r_{sender} \times ID_{sener} \times (1 + Pv_{addressee}) = r_{sender} \times ID_{sender} \times [D + (D \times Pv_{addressee})] = r_{sender} \times ID_{sender} \times (D + Pub_{addressee} = K_2$

Both unsigncryption phase and signcryption phase keys are equivalent to the values they have generated, hence, they are valid to act as an encryption and signature method between entities.

VI.2 Informal Security Analysis

- Proposition 1: The proposed scheme can resist a replay attack.

  Proof: Different current timestamps values were used in the protocol in all transmitted messages, and each message has a maximum transmission delay Δt value. Furthermore, replaying messages does not provide profit to adversary A, since all messages in the authentication phases among all entities must be transmitted within Δt interval. Therefore, the model can resist replay attacks.

- Proposition 2: The proposed scheme can resist a Man-In-The-Middle attack (MITM).

  Proof: Let an adversary A intercept an authentication message with (c,N,Sig,t) and attempt to create a valid key for the signature to bypass the signcryption process. In this case, would require the private key of the addressee to compute the key $K = K_1 + K_2$, where $K_1 = Sig + (N \times Pub_{sender}$ and $K_2 = K_1 \times (1 + Pv_{addressee})$. IPM function is applied for converting the polynomial key to an integer value and creating a valid K_x, completing an unsigncrpytion phase.

  Since the long-term value of the private key is not known to the addressee, adversary A cannot perform a man-in-the-middle attack. Therefore, the model is secured against such attacks.

- Proposition 3: The proposed scheme is secure against various impersonation attacks.

Proof: Let an adversary A try to impersonate a valid communicating entity in the network by creating an authentication message on behalf of that entity. A computes and sends a message to a signscryption sender, such as a general node. Towards generating a valid signscryption and matching the unsigncryption phase of the receiving entity, A requires the calculation of K = $K_1 + K_2$, such that $K_1 = r_{sender} \times ID_{sender} \times D$ and $K_2 = r_{sender} \times ID_{sender} \times (D + Pub_{addressee})$ and both long-term secrets, such as the valid HECC generator,and secret s, and ephemeral secrets, such as random nonces. A is not able to represent the legitimate entity without knowing those secret values, which are not produced in polynomial time by A. The analysis is also valid for the unsigncryption phase and can be applied to all authentication phases that use this method.

As the Batch Authentication Phase (proposition 2), previous knowledge of such secrets by the attacker is required, thus preventing impersonation of KDC or the general node.

The scheme is resilient against impersonation attacks of the cloud server, KDC, fog server, and the General Node, hence, against various impersonation attacks.

- Proposition 4: The proposed scheme can resist an Ephemeral Secret Leakage (ESL) attack.

  Proof: All session keys are calculated via signcryption of two communicating entities, as K = $K_1 + K_2$, such that $K_1 = r_{sender} \times ID_{sender} \times D$ and $K_2 = r_{sender} \times ID_{sender} \times (D + Pub_{addressee})$ for the signcryption phase and $K = K_1 + K_2$, where $K_1 = Sig + (N \times Pub_{sender}$ and $K_2 = K_1 \times (1 + Pv_{addressee})$ for the unsigncryption phase, where $N = h_0(m||K_x||t)$ and $Sig = [(r_{sender} \times ID_{sender}) - (Pv_{sender} \times N)] \times D$. K must undergo an IPM function towards creating (K_x, K_y) keys.

  Adversary A attempts to obtain key K_x, since it is used through in-secure channels. However, the key is a combination of both short- and long-term secrets containing the identities of the communicating entities, HECC generator D, nonces, and timestamps. The session key can be revealed only if A compromises both short-term and long-term secret values. Furthermore, since various nonces and timestamps values are used in the calculation of the session key among several entities, i.e., GN_i, CS_j and KDC_n in all different sessions, even if the session key is revealed for a specific session, other session Keys will not be revealed due to the use of both short-term and long-term secret values.

- Proposition 5: The proposed scheme resists a privileged-insider attack.

  Proof: The privileged-insider user of the trusted authority has registration information on the various entries i.e., $GN_i, CS_j and KDC_n$. The attacker cannot calculate the session key on behalf of a genuine entity, since the key is created with the use of credentials known only by that entity. key K = $K_1 + K_2$, such that $K_1 = r_{sender} \times ID_{sender} \times D$ and $K_2 = r_{sender} \times ID_{sender} \times (D + Pub_{addressee})$ for the signcryption phase and $K = K_1 + K_2$, where $K_1 = Sig + (N \times Pub_{sender}$ and $K_2 = K_1 \times (1 + Pv_{addressee})$ for the unsigncryption phase, where $N = h_0(m||K_x||t)$ and $Sig = [(r_{sender} \times ID_{sender}) - (Pv_{sender} \times N)] \times D$. K must undergo an IPM function towards creating (K_x, K_y) keys. The key is a combination of several long- and short-term secrets not known by the adversary (e.g., private keys, nonces, and timestamps).

According to proposition 3, since A cannot impersonate a valid communicating entity, it cannot compute the session key and, consequently, perform the attack. Therefore, the model resists privileged-insider attacks.

- Proposition 6: The proposed scheme preserves anonymity and untraceability properties.

  Proof: Suppose adversary A can seize any of the messages sent through the communication of the scheme (e.g., c_1,N_1,Sig,t_1), (c_2,N_2,t_2), (Pub_{GN_j},t_{1GN_j})
  and $M_i = (E_{k_{x_{ni}}}, N'_{in}, tg_{n)}$. All messages are computed with the use of random nonce values and timestamps, which help the generation of dynamic and unique messages in different sessions.No information on the identity of any entity is transmitted in a raw format. This idea is present in all message exchange in the protocol, which proves the scheme helps the achievement of anonymity and untraceability.

- Proposition 7: The proposed scheme is resilient against data modification attacks to the cloud server.

  Proof: The cloud server acts as a miner node and receives data from the fog server; therefore, it can prepare and add a block into the existing blockchain through the steps defined in Algorithms 1 and 2. The block is added to the chain and, since a blockchain is a tamper proof technology and attacker A is not able to overtake the required fraction of blocks, it cannot modify the data of a block. Any sudden change to data in the network will mismatch the blockchain ledger, which then warns the scheme of such an event. Therefore, the model is resilient against data modification attacks.

- Proposition 8: The proposed scheme provides data integrity and privacy.

  Proof: The blockchain provides a chain of immutable records. All data are checked prior to being transferred and then stored in the chain; consequently, they are not altered by attackers. Moreover, the use of hash through the protocol for the communication of messages and in the composition of the parameters to be used in the construction of the blockchain provides the collision-resistance property of one-way hash functions. Consequently, obtaining secret information from inside hashed data in the blockchain is computationally infeasible and, as stated in proposition 6, the scheme preserves anonymity and untraceability properties. Therefore, the model provides data integrity and privacy.

- Proposition 9: The proposed scheme provides resistance to eavesdroping attack.

  Proof: During the entire transmission of data (e.g., c_1,N_1,Sig,t_1), (c_2,N_2,t_2), (Pub_{GN_j},$t_{1_{GN_j}}$)
  and $M_i = (E_{k_{x_{ni}}}, N'_{in}, tg_{n)}$, all identities and data are hashed and all messages contain short-term secrets on them (e.g., nonces and timestamps). Consequently, an adversary A that eavesdrops the message is unable to compromise the session, or to steal crucial data that might compromise other sessions. A is also unable to steal vital information such as identities of the communicating entities. Therefore, the protocol is resistant to eavesdrop attacks.

- Proposition 10: The proposed scheme resists Denial of Service (DoS) attack.

Proof: All messages contain short-term parameters such as nonces and timestamps so that the scheme can observe the freshness of the messages during each session. If an adversary A wishes to perform a DoS attack, it must send a valid message to the receiving entity, thus showing it can craft the message.The message would contain a timestamp and random numbers so that the receiving entity can detect how fresh it is and its delay and checks if it is a valid message. If necessary, communication of the sender can be cut. Therefore, the model is resistant to DoS attacks.

- Proposition 11: The proposed scheme provides perfect forward and backward secrecy.

  Proof: Proof: A protocol is classified as able to provide forward and backward secrecy if the exposure of the session key to any instance of the scheme does not compromise the key of previous or future sessions. In our scheme, if adversary A tries to compute session key K = $K_1 + K_2$, such that $K_1 = r_{sender} \times ID_{sender} \times D$ and $K_2 = r_{sender} \times ID_{sender} \times (D + Pub_{addressee})$ for the signcryption phase, or $K = K_1 + K_2$, where $K_1 = Sig + (N \times Pub_{sender}$ and $K_2 = K_1 \times (1 + Pv_{addressee})$ for the unsigncryption one, where $N = h_0(m||K_x||t)$ and $Sig = [(r_{sender} \times ID_{sender}) - (Pv_{sender} \times N)] \times D$, key K must undergo an IPM function towards creating (K_x, K_y) keys. The key contains randomly generated values and secret keys unknown to the adversary and will be different for each session. Therefore, the protocol provides backward and forward secrecy.

- Proposition 12: The proposed scheme is resilient to Sybil attack.

  Proof: On the blockchain consensus side, Sybil attacks are commonly present in blockchain systems based on Proof of Work dynamics. Our protocol uses XRP ledger to sustain the consensus mechanism of the chain; the attack would run a large number of validators, thus convincing other entities they are trustworthy. An attack to XRP ledger [17] is very difficult, since human intervention is necessary for validators to be trusted.

  Regardless of number of validating servers run by a would-be attacker, he/she is not able to influence on what the existing participants consider validated unless those participants choose to trust the attacker's validators. Other servers only listen to the validators they are configured to trusting through either a validator list, or explicit configuration. The attacker must convince targeted humans and businesses to reconfigure their XRP ledger servers for trusting the attacker's validators. Therefore, the attack is highly infeasible without the compromise of other areas of security.

  Each entity uses only one public key and identity and has only one secret key associated per session. All communication is validated by binding via either a signature of the signscryption, or a unique hash produced in the batch authentication phase. Moreover, for a KDC announcement to the general nodes, the attacker, as seen in other propositions, is unable to forge the aforementioned values, thus preventing A from pursuing the attack in a feasible manner. Therefore, the protocol is resilient to Sybil attack.

- Proposition 13: The proposed scheme prevents 51% attack.

Proof: The offending party must control more than 50% of all mining or voting power to perform a 51% attack to the blockchain. XRP Ledger does not use mining in its consensus mechanism [15], in the common sense present in Proof of Work mechanisms. Therefore, the scheme prevents 51% attacks to the chain.

VI.3 Formal Security Validation by AVISPA tool

Our scheme is processed through a formal security of its secrecy and authentication functions on insecure channels. It was validated by Automated Validation of Internet Security Protocols and Applications (AVISPA) [49],a semi-automated validation tool that verifies the security robustness of authentication protocols by checking the secrecy of key parameters and vulnerability to intruders. AVISPA validation is performed through codes written in High-level Protocol Specification Language (HLPSL) [49]. The message exchange of the protocol is translated to an HLPSL code, and each entity is defined as a communication agent that performs roles that contain all the parameters exchanged in the messages (States). Those that must remain secret are signalized and observed during the code execution. If no secret value is vulnerable to be discovered by intruders, the protocol is considered safe. Two of the AVISPA's four security evaluation backends, namely On-the-Fly-Model- checker (OFMC) [8] and Constraint Logic-Based Attack Searcher (CL-AtSe) [50] were used in the validation of the protocol.

All authentication phases performed over an insecure channel were validated. Since all insecure communications are similarly operated over the signcrpytion method, they are equivalent between each other in the validation tool.

Following the Mutual Authentication Phase between the Cloud Server and KDC, Figure 11 displays the role of the cloud server, or signcrypter, and Figure 12 shows the role of the fog server, or the unsigncrypter. Figure 13 depicts the environment section of the code, representing the communication of the declared entities, and Figures 14 and 15 show the protocol successfully provided secrecy and authentication over an insecure channel.

```
role cloudserver(
        CS,KDC: agent,
        Ps,Pu: public_key,
        Add,Sub,Mul,H: hash_func,
        SND, RCV: channel(dy)) played_by CS

def=
        local
                State: nat,
                N,D,IDcsj,T,M,Pv:text,
                K1,K2,C,R,S,S1,K:message
                const s1,s2,s3,s4,s5,auth_al:protocol_id
        init
                State:=0
        transition
                1.State = 0 /\ RCV(start)=|> State':=1 /\N':=new() /\ M':=new()
                /\K1':=Mul(N'.D.IDcsj) /\ K2':=Mul(N'.IDcsj.Add(D.Pu)) /\ K':=Add(K1'.K2')
                /\C':={M'.T}_K' /\  R':=H({M'.K'.T}_K')
                /\S1':=Sub(Mul(N'.IDcsj).Mul(Pv.R')) /\ S':=Mul(D.S1)
                /\SND({M'.T}_K',R'.S'.T)
                /\secret({N'},s1,{CS}) /\secret({IDcsj},s2,{CS})
                /\secret({Pv},s3,{CS}) /\secret({K'},s4,{CS,KDC})
end role
```

Figure 11: HLPSL code of the role of the cloud server in the Mutual Authentication Phase.

```
role fogserver(CS,KDC:agent,
        Ps,Pu:public_key,
        Add,Sub,Mul,HMAC:hash_func,
        SND,RCV:channel(dy))
played_by KDC
def=
        local
                State: nat,
                Pub,M,T:text,
                K1,K2,K,R,S:message
                const s1,s2,s3,s4,s5,auth_al:protocol_id
        init
                State:=0
        transition
                1. State=0/\RCV({M'.T'}_K'.R'.S'.T)=|>
                State':=1
                /\K1':=Add(S'.Mul(R'.Ps))
                /\K2':=Mul(K1'.Add(1.Pub))
                /\K':=Add(K1'.K2')
                /\ secret({Pub},s5,{KDC})
                /\witness(CS,KDC,auth_al,{M'})
end role
```

Figure 12: HLPSL code of the role of the fog server in the Mutual Authentication Phase.

```
role session(CS,KDC:agent,
        Ps,Pu:public_key,
        Add,Sub,Mul,HMAC:hash_func)
def=
        local
                SND1,RCV1,SND2,RCV2:channel(dy)
        composition
                cloudserver(CS,KDC,Ps,Pu,Add,Sub,Mul,HMAC,SND1,RCV1)
                /\fogserver(CS,KDC,Ps,Pu,Add,Sub,Mul,HMAC,SND2,RCV2)
end role

role environment()
 def=
        const
                cloudserver,fogserver:agent,
                ps,pu:public_key,
                add,sub,mul,hmac:hash_func,
                s1,s2,s3,s4,s5,auth_al:protocol_id
        intruder_knowledge={cloudserver,fogserver,ps,pu,add,sub,mul,hmac}
        composition
                session(cloudserver,fogserver,ps,pu,add,sub,mul,hmac)
                /\session(i,fogserver,ps,pu,add,sub,mul,hmac)
                /\session(cloudserver,i,ps,pu,add,sub,mul,hmac)
end role

goal

authentication_on auth_al
secrecy_of s1 secrecy_of s2 secrecy_of s3 secrecy_of s4 secrecy_of s5

end goal

environment()
```

Figure 13: HLPSL code of the enviroment section of the phase.

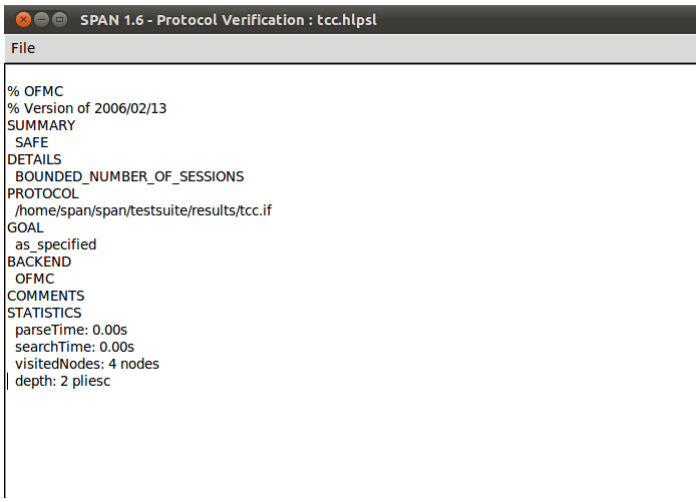

Figure 14: AVISPA result with OFMC backend of BEAP-IoMT.

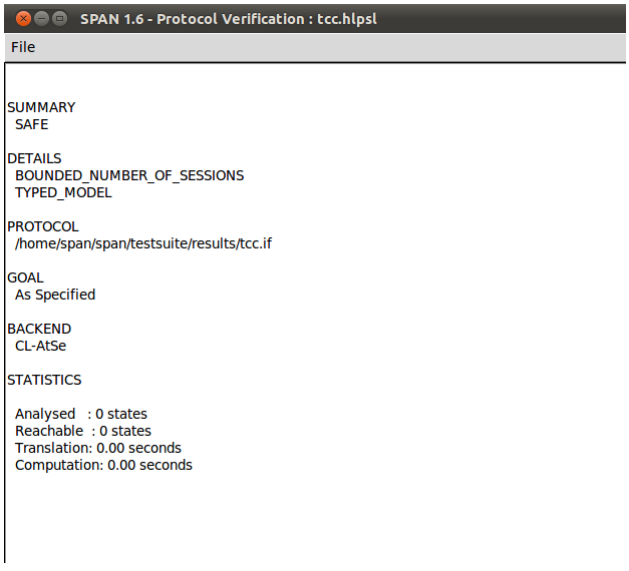

Figure 15: AVISPA result with CL-AtSe backend of BEAP-IoMT.

VI.4 Security and Functionalities Comparison

This section is devoted to a comparison of BEAP-IoMT against other competing schemes regarding functionality and security features. Table 5 shows a comparative analysis of features F1-F18 against those of the schemes of Xu [55] and Bagga [7]. Our protocol provides additional security and functionality features.

Table 5: Comparison of security and functionality features

Feature	**Xu et al.**	**Bagga et al.**	**Proposed scheme**
F1	✓	✓	✓
F2	*X*	✓	✓
F3	✓	✓	✓
F4	*X*	✓	✓
F5	*X*	✓	✓
F6	✓	*X*	✓
F7	*X*	✓	✓
F8	✓	✓	✓
F9	*X*	✓	✓
F10	*X*	✓	✓
F11	✓	*X*	✓
F12	*X*	✓	✓
F13	*X*	*X*	✓
F14	✓	✓	✓
F15	✓	✓	✓
F16	✓	✓	✓
F17	✓	✓	✓
F18	✓	✓	✓

Note: F1: Replay attack protection; F2: Man-In-The-Middle attack protection; F3: Identity impersonation attack protection; F4: ESL attack protection; F5: Privileged-insider attack protection; F6: Provides anonymity and untraceability properties; F7: Protection against data modification attack; F8: Provides data integrity and privacy; F9: Eavesdropping attack protection; F10: DoS attack protection; F11: Provides perfect backward and forward secrecy; F12: Sybil attack protection; F13: 51% attack protection; F14: Dynamic node addition phase; F15: Blockchain-enabled solution; F16: Formal security verification by a validation tool; F17: Group key management phase; F18: CK threat model implementation. ✓: Scheme supports mentioned feature; *X*: Feature is not supported or addressed.

VII PERFORMANCE ANALYSIS

This section addresses a comparative analysis of several performance parameters of the protocol and a comparison to competing schemes developed by Bagga [7] and Xu [55]. The categories presented are computational and communication costs of the several phases of the protocol. A study on the initial storage overhead required for the entities in our network model and a brief comparison of the security functionalities of the schemes are provided. The computational power of the protocol was checked towards a proper performance analysis. Symbols T_{enc}, T_{dec}, T_{hpm}, T_{hpa}, T_{pm}, T_{pa}, and T_h stand for processing times involved in encryption, decryption, HECC point multiplication, HECC point addition, HECC point multiplication, HECC point addition, integer pairing, and hash operations, respectively.

The computational cost comparison was split into the following three segments: single authentication between entities, which includes mutual authentication of KDC and cloud server and authentication between the general nodes of the group, batch authentication of the group, and group key management.

As addressed elsewhere, CSs, KDCs, GNs, and TA comprise the network model of the scheme. Cloud and fog servers have large computational power and sufficient amount of storage capacity, whereas the generic nodes are more resource-constrained. Moreover, TA does not act directly on the authentication process of the protocol and is mostly used as a registration tool.

VII.1 Computational Costs

The computational cost calculation is based on the processing time required for each operation of the protocol and the number of operations necessary for its implementation. Table 7 shows the average execution time of operations by all entities in the respective authentication step for the computational overhead analysis of the schemes. A comparison of the results with those from Bagga [7] and Xu [55] is provided in Table 8.

The average execution time used was taken from Ostad-Sharif [43], as shown in Table 6; the values were obtained from an LG G6 smartphone using OpenSSL library and Bagga [7]. Regarding the hash function, the difference in execution time between SHA256 and SHA1 algorithms is negligible, and, therefore, both will be referred to as T_h under SHA256 execution time.

VII.2 Communication Costs

The communication cost calculation is based on the number of bits sent for each message produced by the protocol and the number of messages required for its operation and enables an evaluation of the bandwidth to be demanded. It considers an identity, a random number (nonce/secret/message), a timestamp, a curve point of form $P = (P_x, P_y)$, where x and y are coordinates of a point P in the HECC curve, and $h(.)$ and $h_1(.)$ are hash digest of SHA-256 and SHA-1, with sizes 160, 160, 64, (192 + 192), 256, 160 bits, respectively. Table 9 shows the calculation and results of the communication cost of each phase of the authentication procedure.

Table 10 provides the communication costs of the schemes of Bagga [7] and Xu [55], using the values

Table 6: Average execution time

Cryptographic element	**Time**	**Description**
$T_{ECC_{pm}}$	537.033 μs	*160-bit elliptic curve point multiplication*
$T_{ECC_{pa}}$	2.237 μs	*160-bit elliptic curve point addition*
$T_{HECC_{pm}}$	609.419 μs	*160-bit hash function*
$T_{HECC_{pa}}$	2.539 μs	*256-bit hash function*
T_{symenc}	111.126 ns	*AES-128 symmetric encryption algorithm*
T_{symdec}	111.126 ns	*AES-128 symmetric decryption algorithm*
$T_{192-symenc}$	129.668 ns	*AES-192 symmetric encryption algorithm*
$T_{192-symdec}$	129.668 ns	*AES-192 symmetric decryption algorithm*
T_{bp}	2.5504128 ms	*Bilinear pairing*
T_h	0.121 μs	*One-way hash function*

Table 7: Computational cost of BEAP-IoMT (in milliseconds)

Mutual Authentication between Cloud Server and KDC	
Entity	**Computational Cost**
Cloud Server	$T_{HECC_{pa}} + 2T_{HECC_{pm}} +$ $\mathrm{T}_{192-symenc} + 2T_h + T_{192-symdec} \approx 1,222$
KDC	$T_{192-symdec} + 2T_h +$ $\mathrm{T}_{192-symenc} + 2T_{HECC_{pm}} + T_{HECC_{pa}} \approx$ 1,222
General Node Authentication Phase	
Entity	**Computational Cost**
General Node	$n - 1(2T_{HECC_{pa}} + 4T_{HECC_{pm}} +$ $2\mathrm{T}_{192-symenc} + T_{192-symdec} + 3T_h) \approx$ 2,4435n - 2,4435
Batch Authentication Phase	
Entity	**Computational Cost**
KDC	$n(T_{192-symenc}) + 2T_{192-symdec} + 2T_h +$ $2\mathrm{T}_{HECC_{pm}} + T_{HECC_{pa}} \approx 1,222n$
General Node	$n(2T_h + 2T_{HECC_{pm}} + T_{HECC_{pa}} +$ $2\mathrm{T}_{192-symenc}) + T_{192-symdec} \approx 1,222n$
Group Key Generation Phase	
Entity	**Computational Cost**
KDC	$T_h + n(t_{192-symenc}) \approx 0,121 + 0,130n$
General Node	$n(T_h + t_{192-symdec}) \approx 0,251n$
Cloud Server	$T_h + T_{192-symdec} \approx 0,251$

offered on their schemes, respectively. The total cost of our protocol and a comparison among the schemes are shown in Table 10.

Table 8: Comparison of computational costs

Scheme	Computational Cost (in milliseconds)
Xu et. al.	$(n+7)T_{ECC_{pm}} + 4T_{bp} + (n+3)T_h + T_{symenc} + T_{symdec} + (n+1)T_{ECC_{pa}}$ $\approx 13,962 + 0,539n$
Bagga et al.	$(6n+2)T_h + (8n+1)T_{ECC_{pa}} + (12n+1)T_{ECC_{pm}} + nT_{enc} + nT_{dec} + 3T_{bp}$ $\approx 8,189 + 6,463n$
BEAP-IoMT	$(6n+2)T_{HECC_{pm}} + (3n+1)T_{HECC_{pa}} + (5n+1)T_{192-symenc}+$ (5n + 2)$T_{192-symdec} + (5n+3)T_h \approx 1,222 + 3,667n$

Table 9: Communication cost of BEAP-IoMT (in bits)

Mutual Authentication between Cloud Server and KDC	
Entity	**Communication Cost**
Cloud Server	$(C + N + Sign + TS) = ((192 + 160 + 64)+$ (160+192+64+256) +(672) + (64)) = 1824
KDC	$(C + N + TS) = ((192 + 160 + 64)+$ (160+192+64+256) + (64)) = 1152
General Node Authentication Phase	
Entity	**Communication Cost**
General Node	Sent: $n(Pub_{GN} + TS) + n - 1(C + N + Sign$ + TS) = 256n + 1824n - 1824 = 2080n - 1824 Received: n-1(Pub$_{GN} + TS)+$ n-1(C + N + TS) = 1408n - 1408
Batch Authentication Phase	
Entity	**Communication Cost**
KDC	$Sent : n(ID_{KDC} + Pub_{KDC} + m + TS) =$ n(160 + 192 + 416 + 64) = 832n Received: 2304n
General Node	$Sent : n(m_{in}^{'} + C + N + Sign + 2TS) =$ n(352 + 416 + 672 + 672 + 128) = 2304n Received: 832
Group Key Generation Phase	
Entity	**Communication Cost**
KDC	$(Sent : n(ENC_{kx} + N + TS) + (C_{GK} + TS) =$ n((160n + 320 + 416 + 192) + 672 + 64) + (160n + 992 + 64) = 160n^2 + 1824n + 1056
General Node	$Received : n(160n + 1664)$
Cloud Server	$Received : 160n + 1056$

Table 10: Comparison of communication costs

Scheme	Communication Cost (in bits)
Xu et al.	$Sent : (5C + 2T)n = (5.256 + 2.64)n = 1408n$ Received: $((3n^2 - n)C + n^2T) = 832n^2 - 256n$
Bagga et al.	$Sent : 5664n$ Received: -
BEAP-IoMT	$Sent : 1824 + 1152 + (2080n - 1824) + 832n +$ $1984n + 160n^2 + 1824n + 1056 =$ $160n^2 + 6720n + 2208$ Received: (1824 + 1152) + (1408n - 1408) + (2304n) + (832n) + n(160n + 1664) + (160n + 1056) = $160\ n^2 + 6368n + 2624$

VII.3 Storage Overhead

We analyze the storage overheads of the proposed scheme and the competing protocols, Xu [55] and Bagga [7], by calculating the amount of storage required by each entity during the initial registration phase.

In the following analysis, the storage overhead is calculated as the amount of storage required by each of the main entities of the scheme (namely, KDC, GN, CS). Parameters such as hash functions and encryption algorithms were assumed to show negligible weight in terms of storage size. The results are shown in Table 11. Analysing the results, we can see that both Xu and Bagga, considers two entities in their storage overhead calculation (namely, KDC and GN for Xu, and Vehicle and RSU for Bagga), while our protocol takes three devices into consideration, this results in the proposed scheme requiring more storage overhead. However, we can see that our overall cost of 2336 bits is lesser than Bagga's 2720 bits. Therefore, considering that our scheme considers a larger amount of devices being registered in the network, our results are comparable to the competition.

VIII CONCLUSIONS

The wider applications of IoT environments due to the increasing uses of smart technologies have been expanded to several industrial sectors, including e-health/m-health services, which can provide easier and more advanced assistance to a much larger user base, especially to those with limited mobility and who reside in areas of difficult access. Moreover, such applications have yielded smarter solutions to sectors that require growing, faster, and efficient ones.

The BEAP-IoMT protocol can be used in the development of efficient and safe IoMT systems, or in other IoT scenarios for protecting patient data and providing more secure and faster services. It aims to deliver security through the use of advanced cryptographic techniques, blockchain, and efficient communication.

Regarding its security features, it shows additional support in comparison to other schemes. Xu [55]

Table 11: Comparison of storage overhead (in bits)

Xu et. al.	
Entity	**Storage Overhead**
KDC	$(P_{pub} + s) = (160 + 160) = 320$
GN	$(S_i + ID_i + W_i + A_i) =$ (320 + 160 + 320 + 160) = 960
Bagga et. al.	
Entity	**Storage Overhead**
Vehicle	1600
RSU	1120
BEAP-IoMT	
Entity	**Storage Overhead**
Cloud Server	$(RID_{CS_j} + RID_{TA} + Pub_{CS_j} + Pv_{CS_j}) =$ (320 + 320 + 192 + 160) = 992
KDC	$(RID_{KDC_n} + Pub_{KDC_n} + Pv_{KDC_n})$ (320 + 192 + 160) = 672
GN	$(RID_{GN_i} + Pub_{GN_i} + Pv_{GN_i})$ (320 + 192 + 160) = 672

addressed no features such as perfect backward secrecy and counter-measures to attacks (e.g., Privileged-insider, MitM, ESL, eavesdropping, DoS, Sybil, and data modifications) and anonymity and untraceability properties and an analysis of the possibility of a 51% attack to the blockchain were not discussed in Bagga [7].

Despite the aforementioned differences, the computational cost of our protocol was lower in comparison to that of Bagga [7] and comparable to that of Xu [55]. Moreover, our scheme involves more entities than that of Xu [55], since KDC is considered not fully trusted. Its communication costs are lower than those of [55] and comparable to the ones of [7].As addressed elsewhere, it provides increased security features.

Such results are a consequence of the use of signcryption and hybrid cryptography, which led to an overall performance comparable to that of the competing literature and better security features in the IoT scenario.

Future studies will include a proper implementation of the scheme in an experimental setup and additional security objectives such as physical capture of devices.

IV CONCLUSIONS

The main objective of this research was the development of a new authentication protocol for IoMT communication that uses blockchain. Therefore, BEAP-IoMT protocol has been proposed towards communication for IoT, m-health, considering 3GPP architecture and Blockchain. The protocol focused on the creation of a Blockchain Enabled Authentication Protocol (BEAP-IoMT) using group and batch authentication techniques for improving performance and making the communication of the network more scalabe and efficient. The scheme also employs advanced cryptographic tools such as hyper elliptic curves, and signcryption for enhancing its key creation and exchange in both security and performance, as well as fog servers as its key distribution centers for handling the large workload and processing of the authentication of the many devices belonging to the network towards delivering faster response times. Lastly, the protocol uses 5G and its features as the backbone of its architecture for providing the expected results. Its security and performance were evaluated and compared to other proposals. The security evaluation regarded the fulfillment of properties such as confidentiality, integrity, privacy, anonymity, untraceability, perfect forward and backward secrecy, and resistance to several attacks(e.g., man-in-the- middle, replay, data modification, sybil, 51%, among others). It has proved more robust than the other proposals, delivering more security and functionality features. The performance evaluation consisted of the measurement of computational, communication, and storage costs. The computational cost was assessed according to the computational time of operations necessary for each authentication phase, whereas the communication cost was measured in bits, taking into account all parameters present in all messages exchanged among entities during each authentication phase. The storage cost, measured in bits, was obtained by calculating the overhead required by each entity during the initial registration phase for its storage.

The performance comparison with the competing literature showed very comparable results; however, it must be observed that our architecture considers an increase in the number of entities present in the network (cloud servers, general nodes and key distribution centers). Moreover, the proposed protocol added security and functionality features and, as such, can be assessed as a better performing authentication mechanism than those of Xu [55] and Bagga [7].

The scheme was validated by AVISPA and proved secure for use, taking into account the described attack models and system architecture.

References

[1] Shubhani Aggarwal, Neeraj Kumar, and Prosanta Gope. “An Efficient Blockchain-Based Authentication Scheme for Energy-Trading in V2G Networks”. In: *IEEE Transactions on Industrial Informatics* 17.10 (2021), pp. 6971–6980. DOI: 10.1109/TII.2020.3030949.

[2] Seyed Farhad Aghili et al. “Closed-loop and open-loop authentication protocols for blockchain-based IoT systems”. In: *Information Processing Management* 58.4 (2021), p. 102568. ISSN: 0306-4573. DOI: https://doi.org/10.1016/j.ipm.2021.102568. URL: https://www.sciencedirect.com/science/article/pii/S0306457321000698.

[3] Raluca Maria Aileni and George Suciu. “IoMT: A Blockchain Perspective”. In: *Decentralised Internet of Things: A Blockchain Perspective*. Ed. by Mohammad Ayoub Khan et al. Cham: Springer International Publishing, 2020, pp. 199–215. ISBN: 978-3-030-38677-1. DOI: 10.1007/978-3-030-38677-1_9. URL: https://doi.org/10.1007/978-3-030-38677-1_9.

[4] Raifa Akkaoui. “Blockchain for the Management of Internet of Things Devices in the Medical Industry”. In: *IEEE Transactions on Engineering Management* (2021), pp. 1–12. DOI: 10.1109/TEM.2021.3097117.

[5] *ants.dvi*. https://crypto.stanford.edu/~dabo/pubs/papers/DDH.pdf. (Accessed on 02/23/2023).

[6] L. M. Bach, B. Mihaljevic, and M. Zagar. “Comparative analysis of blockchain consensus algorithms”. In: *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*. 2018, pp. 1545–1550. DOI: 10.23919/MIPRO.2018.8400278.

[7] Palak Bagga et al. “Blockchain-based batch authentication protocol for Internet of Vehicles”. In: *Journal of Systems Architecture* 113 (2021), p. 101877. ISSN: 1383-7621. DOI: https://doi.org/10.1016/j.sysarc.2020.101877. URL: https://www.sciencedirect.com/science/article/pii/S1383762120301569.

[8] David Basin, Sebastian Mödersheim, and Luca Vigano. “OFMC: A symbolic model checker for security protocols”. In: *International Journal of Information Security* 4.3 (2005), pp. 181–208.

[9] David Basin et al. “A Formal Analysis of 5G Authentication”. In: *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*. CCS ’18. Toronto, Canada: Association for Computing Machinery, 2018, pp. 1383–1396. ISBN: 9781450356930. DOI: 10.1145/3243734.3243846. URL: https://doi.org/10.1145/3243734.3243846.

[10] Nursena Baygin, Mehmet Baygin, and Mehmet Karakose. “Blockchain Technology: Applications, Benefits and Challenges”. In: *2019 1st International Informatics and Software Engineering Conference (UBMYK)*. 2019, pp. 1–5. DOI: 10.1109/UBMYK48245.2019.8965565.

[11] Bharat Bhushan et al. “Untangling blockchain technology: A survey on state of the art, security threats, privacy services, applications and future research directions”. In: *Computers Electrical Engineering* 90 (2021), p. 106897. ISSN: 0045-7906. DOI: https://doi.org/10.1016/j.compeleceng.2020.106897. URL: https://www.sciencedirect.com/science/article/pii/S0045790620307497.

[12] Ravishankar Borgaonkar et al. “New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols”. In: *Proceedings on Privacy Enhancing Technologies* 2019 (July 2019), pp. 108–127. DOI: 10.2478/popets-2019-0039.

[13] Ravishankar Borgaonkar et al. “New Privacy Threat on 3G, 4G, and Upcoming 5G AKA Protocols”. In: *Proceedings on Privacy Enhancing Technologies* 2019 (July 2019), pp. 108–127. DOI: 10.2478/popets-2019-0039.

[14] Claudia Campolo et al. “Towards 5G Network Slicing for the V2X Ecosystem”. In: June 2018, pp. 400–405. DOI: 10.1109/NETSOFT.2018.8459911.

[15] Ran Canetti and Hugo Krawczyk. “Analysis of Key-Exchange Protocols and Their Use for Building Secure Channels”. In: *Advances in Cryptology — EUROCRYPT 2001*. Ed. by Birgit Pfitzmann. Berlin, Heidelberg: Springer Berlin Heidelberg, 2001, pp. 453–474. ISBN: 978-3-540-44987-4.

[16] Ran Canetti and Hugo Krawczyk. *Universally Composable Notions of Key Exchange and Secure Channels*. Cryptology ePrint Archive, Paper 2002/059. https://eprint.iacr.org/2002/059. 2002. URL: https://eprint.iacr.org/2002/059.

[17] Brad Chase and Ethan MacBrough. *Analysis of the XRP Ledger Consensus Protocol*. 2018. DOI: 10.48550/ARXIV.1802.07242. URL: https://arxiv.org/abs/1802.07242.

[18] Qingnan Chen et al. “An Identity-Based Cross-Domain Authenticated Asymmetric Group Key Agreement”. In: *Information* 12.3 (2021). ISSN: 2078-2489. DOI: 10.3390/info12030112. URL: https://www.mdpi.com/2078-2489/12/3/112.

[19] Malathi Devarajan and N. Sasikaladevi. “An hyper elliptic curve based efficient signcryption scheme for user authentication”. In: *Journal of Intelligent & Fuzzy Systems* 39 (2020). 6, pp. 8487–8498. ISSN: 1875-8967. DOI: 10.3233/JIFS-189166. URL: https://doi.org/10.3233/JIFS-189166.

[20] Nimra Dilawar et al. “Blockchain: Securing Internet of Medical Things (IoMT)”. In: *International Journal of Advanced Computer Science and Applications* 10.1 (2019). DOI: 10.14569/IJACSA.2019.0100110. URL: http://dx.doi.org/10.14569/IJACSA.2019.0100110.

[21] Danny Dolev and Andrew Chi-Chih Yao. “On the security of public key protocols”. In: *22nd Annual Symposium on Foundations of Computer Science (sfcs 1981)* (1983), pp. 350–357.

[22] Michael J. Fischer. “The consensus problem in unreliable distributed systems (a brief survey)”. In: *Foundations of Computation Theory*. Ed. by Marek Karpinski. Berlin, Heidelberg: Springer Berlin Heidelberg, 1983, pp. 127–140. ISBN: 978-3-540-38682-7.

[23] Filippos Fotopoulos et al. “A Blockchain-enabled Architecture for IoMT Device Authentication”. In: *2020 IEEE Eurasia Conference on IOT, Communication and Engineering (ECICE)* (2020), pp. 89–92.

[24] Ramachandran Ganesan, Mohan Gobi, and Kanniappan Vivekanandan. “A Novel Digital Envelope Approach for A Secure E-Commerce Channel”. In: *Int. J. Netw. Secur.* 11 (2010), pp. 121–127.

[25] Neha Garg et al. “BAKMP-IoMT: Design of Blockchain Enabled Authenticated Key Management Protocol for Internet of Medical Things Deployment”. In: *IEEE Access* 8 (2020), pp. 95956–95977. DOI: 10.1109/ACCESS.2020.2995917.

[26] Pierrick Gaudry. “An Algorithm for Solving the Discrete Log Problem on Hyperelliptic Curves”. In: *Advances in Cryptology — EUROCRYPT 2000*. Ed. by Bart Preneel. Berlin, Heidelberg: Springer Berlin Heidelberg, 2000, pp. 19–34. ISBN: 978-3-540-45539-4.

[27] Jyoti Grover. “Security of Vehicular Ad Hoc Networks using blockchain: A comprehensive review”. In: *Vehicular Communications* 34 (2022), p. 100458. ISSN: 2214-2096. DOI: https://doi.org/10.1016/j.vehcom.2022.100458. URL: https://www.sciencedirect.com/science/article/pii/S2214209622000055.

[28] Yihao Guo, Zhiguo Wan, and Xiuzhen Cheng. "When blockchain meets smart grids: A comprehensive survey". In: *High-Confidence Computing* 2.2 (2022), p. 100059. ISSN: 2667-2952. DOI: https://doi.org/10.1016/j.hcc.2022.100059. URL: https://www.sciencedirect.com/science/article/pii/S2667295222000113.

[29] S Haber. "WS, 1991."How to time-stamp a digital document"". In: *Journal of Cryptology* (), pp. 99–111.

[30] Kashif Hameed et al. "Integration of 5G and Block-Chain Technologies in Smart Telemedicine Using IoT". In: *Journal of Healthcare Engineering* 2021 (Mar. 2021), p. 8814364. ISSN: 2040-2295. DOI: 10.1155/2021/8814364. URL: https://doi.org/10.1155/2021/8814364.

[31] Dongyan Huang, Xiaoli Ma, and Shengli Zhang. "Performance Analysis of the Raft Consensus Algorithm for Private Blockchains". In: *IEEE Transactions on Systems, Man, and Cybernetics: Systems* 50.1 (2020), pp. 172–181. DOI: 10.1109/TSMC.2019.2895471.

[32] Huawei Huang et al. *When Blockchain Meets Distributed File Systems: An Overview, Challenges, and Open Issues*. Mar. 2020. DOI: 10.1109/ACCESS.2020.2979881.

[33] Srinivas Jangirala, Ashok Kumar Das, and Athanasios V. Vasilakos. "Designing Secure Lightweight Blockchain-Enabled RFID-Based Authentication Protocol for Supply Chains in 5G Mobile Edge Computing Environment". In: *IEEE Transactions on Industrial Informatics* 16.11 (2020), pp. 7081–7093. DOI: 10.1109/TII.2019.2942389.

[34] Deng Jian-zhi, Cheng Xiao-hui, and Gui Qiong. "Design of Hyper Elliptic Curve Digital Signature". In: *2009 International Conference on Information Technology and Computer Science*. Vol. 2. 2009, pp. 45–47. DOI: 10.1109/ITCS.2009.146.

[35] Li Jiang et al. "Blockchain Empowered Wireless Power Transfer for Green and Secure Internet of Things". In: *IEEE Network* 33.6 (2019), pp. 164–171. DOI: 10.1109/MNET.001.1900008.

[36] Li Jiang et al. "Incentivizing Resource Cooperation for Blockchain Empowered Wireless Power Transfer in UAV Networks". In: *IEEE Transactions on Vehicular Technology* 69.12 (2020), pp. 15828–15841. DOI: 10.1109/TVT.2020.3036056.

[37] Junbin Kang et al. "An ultra light weight and secure RFID batch authentication scheme for IoMT". In: *Computer Communications* 167 (2021), pp. 48–54. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2020.12.004. URL: https://www.sciencedirect.com/science/article/pii/S0140366420320107.

[38] Umair Khalid et al. "A decentralized lightweight blockchain-based authentication mechanism for IoT systems". In: *Cluster Computing* 23.3 (Sept. 2020), pp. 2067–2087. ISSN: 1573-7543. DOI: 10.1007/s10586-020-03058-6. URL: https://doi.org/10.1007/s10586-020-03058-6.

[39] Leslie Lamport, Robert Shostak, and Marshall Pease. "The Byzantine Generals Problem". In: *ACM Trans. Program. Lang. Syst.* 4.3 (July 1982), pp. 382–401. ISSN: 0164-0925. DOI: 10.1145/357172.357176. URL: https://doi.org/10.1145/357172.357176.

[40] Kristin Lauter and Katherine Stange. “The Elliptic Curve Discrete Logarithm Problem and Equivalent Hard Problems for Elliptic Divisibility Sequences”. In: vol. 5381. Apr. 2008. ISBN: 978-3-642-04158-7. DOI: 10.1007/978-3-642-04159-4_20.

[41] Ying-Chang Liang. “Blockchain for Dynamic Spectrum Management”. In: Jan. 2020, pp. 121–146. ISBN: 978-981-15-0775-5. DOI: 10.1007/978-981-15-0776-2_5.

[42] Oscar Novo. “Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT”. In: *IEEE Internet of Things Journal* 5.2 (2018), pp. 1184–1195. DOI: 10.1109/JIOT.2018.2812239.

[43] Arezou Ostad-Sharif, Dariush Abbasinezhad-Mood, and Morteza Nikooghadam. “Efficient utilization of elliptic curve cryptography in design of a three-factor authentication protocol for satellite communications”. In: *Computer Communications* 147 (2019), pp. 85–97. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2019.08.018. URL: https://www.sciencedirect.com/science/article/pii/S0140366418310235.

[44] Panos Papadimitratos, Virgil Gligor, and Jean-Pierre Hubaux. “Securing Vehicular Communications - Assumptions, Requirements, and Principles”. In: *4th Workshop Embedded Security in Cars* (Jan. 2006).

[45] Mayuresh Sunil Pardeshi, Ruey-Kai Sheu, and Shyan-Ming Yuan. “Hash-Chain Fog/Edge: A Mode-Based Hash-Chain for Secured Mutual Authentication Protocol Using Zero-Knowledge Proofs in Fog/Edge”. In: *Sensors* 22.2 (2022). ISSN: 1424-8220. DOI: 10.3390/s22020607. URL: https://www.mdpi.com/1424-8220/22/2/607.

[46] Filippos Pelekoudas-Oikonomou et al. “Blockchain-Based Security Mechanisms for IoMT Edge Networks in IoMT-Based Healthcare Monitoring Systems”. In: *Sensors* 22.7 (2022). ISSN: 1424-8220. DOI: 10.3390/s22072449. URL: https://www.mdpi.com/1424-8220/22/7/2449.

[47] Mohammad Saidur Rahman, Abdulatif Alabdulatif, and Ibrahim Khalil. “Privacy Aware Internet of Medical Things Data Certification Framework on Healthcare Blockchain of 5G Edge”. In: *Computer Communications* 192 (2022), pp. 373–381. ISSN: 0140-3664. DOI: https://doi.org/10.1016/j.comcom.2022.06.013. URL: https://www.sciencedirect.com/science/article/pii/S0140366422002158.

[48] William Stallings. *Cryptography and Network Security: Principles and Practice*. 3rd. Pearson Education, 2002. ISBN: 0130914290.

[49] *The AVISPA Project: European Union in the Future and Emerging Technologies (FET Open)*. URL: https://www.avispa-project.org/.

[50] Mathieu Turuani. “The CL-Atse protocol analyser”. In: *International conference on rewriting techniques and applications*. Springer. 2006, pp. 277–286.

[51] Jing Wang et al. “Blockchain-Based Anonymous Authentication With Key Management for Smart Grid Edge Computing Infrastructure”. In: *IEEE Transactions on Industrial Informatics* 16.3 (2020), pp. 1984–1992. DOI: 10.1109/TII.2019.2936278.

[52] Qianwen Wang et al. “A Comparative Study of Blockchain Consensus Algorithms”. In: *Journal of Physics: Conference Series* 1437.1 (Jan. 2020), p. 012007. DOI: 10.1088/1742-6596/1437/1/012007. URL: https://dx.doi.org/10.1088/1742-6596/1437/1/012007.

[53] Zhihao Wang, Ru Huo, and Shuo Wang. “A Lightweight Certificateless Group Key Agreement Method without Pairing Based on Blockchain for Smart Grid”. In: *Future Internet* 14.4 (2022). ISSN: 1999-5903. DOI: 10.3390/fi14040119. URL: https://www.mdpi.com/1999-5903/14/4/119.

[54] Mohammad Wazid et al. “Security in 5G-Enabled Internet of Things Communication: Issues, Challenges, and Future Research Roadmap”. In: *IEEE Access* 9 (2021), pp. 4466–4489. DOI: 10.1109/ACCESS.2020.3047895.

[55] Zisang Xu et al. “A Blockchain-Based Authentication and Dynamic Group Key Agreement Protocol”. In: *Sensors* 20.17 (2020). ISSN: 1424-8220. DOI: 10.3390/s20174835. URL: https://www.mdpi.com/1424-8220/20/17/4835.

[56] Hongwei Zhang, Jinsong Wang, and Yuemin Ding. “Blockchain-based decentralized and secure keyless signature scheme for smart grid”. In: *Energy* 180 (2019), pp. 955–967. ISSN: 0360-5442. DOI: https://doi.org/10.1016/j.energy.2019.05.127. URL: https://www.sciencedirect.com/science/article/pii/S0360544219310096.